IMPLEMENTASI ALGORITMA *BLOWFISH* DAN METODE *LEAST SIGNIFICANT BIT INSERTION* PADA VIDEO MP4

Dedy Abdullah[1], Doni Nugroho Saputro[2]

[1,2]*Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu*
Jl. Bali PO BOX 118 Telp (0736)227665' FAX (0736)26161, Bengkulu 38119

[1]dedy_abdullah@umb.ac.id
[2]donins2612@gmail.com

Abstrak : Penggunaan teknologi komputer memudahkan manusia dalam membuat dan menggandakan karya-karya multimedia seperti lagu, musik, gambar, dan video. Salah satu format video yang populer yang diunggah di internet adalah berkas video mp4. Banyaknya video mp4 yang beredar di internet dapat kita gunakan untuk menyisipkan pesan tanpa diketahui oleh orang lain. Salah satu cara untuk memanfaatkan berkas video adalah dengan menggunakan teknik video steganografi. Metode yang digunakan untuk steganografi dalam penelitian adalah *Low Bit Encoding* dengan enkripsi algoritma *Blowfish*. Metode *Low Bit Encoding* digunakan untuk menyisipkan pesan kedalam berkas video pada setiap bit yang paling tidak berpengaruh sedangkan algoritma *Blowfish* digunakan sebagai pengamanan agar pesan yang disisipkan kedalam berkas video tidak dapat dibaca oleh orang yang tidak berhak. Hasil penelitian menunjukkan bahwa penggunaan metode *Low Bit Encoding* dan Algoritma *Blowfish* pada video berformat mp4 berhasil dan tidak memperlihatkan perbedaan yang signifikan terhadap kualitas video aslinya.
Kata kunci : Video mp4, *Low Bit Encoding, Least Significant Bit, Blowfish*.

Abstract: The use of computer technology enable people to create and multiply multimedia works such as song , music , pictures , and video . One popular video format that is uploaded on the internet is mp4. Many mp4 video circulating on the internet can be used to insert the message without being noticed by others. One way to take advantage of video files is to use video steganography techniques. The method used for steganography in this research is Low Bit Encoding with Blowfish encryption algorithm. Low Bit Encoding method used to insert a message into a video file at least every bit as influential while the Blowfish algorithm is used as security for a message that is inserted into the video file can not be read by unauthorized people. The results showed that the use of methods Low Bit Encoding and Blowfish algorithm in mp4 format video works and does not show a significant difference to the quality of the video.
Keywords: Steganography, mp4, Low Bit Encoding, Least Significant Bit, Blowfish

I. PENDAHULUAN

Penggunaan teknologi komputer memudahkan manusia dalam membuat dan menggandakan karya-karya multimedia seperti lagu, musik, gambar, dan video. Salah satu format video yang populer yang diunggah di internet adalah berkas video mp4. Banyaknya video mp4 yang beredar di internet dapat kita gunakan untuk menyisipkan pesan tanpa diketahui oleh orang lain. Salah satu cara untuk memanfaatkan berkas video adalah dengan menggunakan teknik video steganografi.

Steganografi merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi "rahasia" di dalam informasi lainnya. Steganografi mempunyai sejarah yang hampir sama dengan kriptografi,

keduanya banyak digunakan ketika zaman perang [1]. Metode steganografi yang digunakan adalah *Least Significant Bit Insertion* (LSB). Pengunaan metode ini populer karena implementasinya yang sederhana dan dapat disisipkan informasi yang lebih banyak dibanding *Spread Spectrum method* , *Redundant Pattern Encoding*. Namun pada perkembangannya diketahui bahwa pengunaan metode *Least Significant Bit Insertion* pada video steganografi rentan terhadap serangan analisis statistik dan proses steganalisis.

Untuk menjaga keamanan data pada proses video steganografi menggunakan metode *Least Significant Bit Insertion* maka penulis menggunakan teknik enkripsi terlebih dahalu terhadap informasi yang akan disisipkan. Motede enkripsi yang digunakan adalah metode *Blowfish* karena metode ini memiliki tingkat efisiensi yang lebih tinggi dibandingkan metode lain seperti AES, DES, dan lain-lain

II. DASAR TEORI

2.1. Kriptografi

Kriptografi berasal dari bahasa Yunani, menurut bahasa dibagi menjadi menjadi dua *kripto* dan *graphia*, *kripto* berarti "*secret*" (rahasia) dan *graphia* berarti "*writing*" (tulisan). Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain [1].

Jika anda bertukar pesan (misalnya surat) dengan orang lain, maka anda tentu ingin pesan yang anda kirim sampai ke pihak yang dituju dengan aman [2].

Ada beberapa istilah-istilah yang penting dalam kriptografi, yaitu :

1. Pesan (*Plaintext* dan *Ciphertext*) : Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Pesan asli disebut plainteks (*plaintext*) atau teks-jelas (*cleartext*). Sedangkan pesan yang sudah disandikan disebut cipherteks (*chipertext*).
2. Pengirim dan Penerima : Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan.
3. Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan.
4. Kriptanalisis dan Kriptologi: Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan chiperteks menjadi plainteks tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalis. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisis.
5. Enkripsi dan Dekripsi : Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering*. Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*) atau *deciphering*.
6. *Cipher* dan Kunci : Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enchipering* dan *dechipering*, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Kunci (*key*) adalah parameter yang digunakan untuk transformasi *enciphering* dan *dechipering*. Kunci biasanya berupa *string* atau deretan bilangan.

2.2. STEGANOGRAFI

Steganografi merupakan seni komunikasi rahasia dengan menyembunyikan pesan pada objek yang tampaknya tidak berbahaya. Keberadaan pesan steganografi adalah rahasia. Istilah Yunani ini berasal dari kata *Steganos*, yang berarti tertutup dan *Graphia*, yang berarti menulis [3].

Steganografi adalah jenis komunikasi yang tersembunyi, yang secara harfiah berarti "tulisan tertutup". Pesannya terbuka, selalu terlihat, tetapi tidak terdeteksi bahwa adanya pesan rahasia. Deskripsi lain yang populer untuk steganografi adalah *Hidden in Plain Sight* yang artinya tersembunyi di depan mata. Sebaliknya, kriptografi adalah tempat pesan acak, tak dapat dibaca dan keberadaan pesan sering dikenal [4].

Istilah steganografi berasal dari bahasa Yunani, yaitu *steganos* yang berarti penyamaran atau penyembunyian dan *graphein* yang berarti tulisan. Jadi, steganografi bisa diartikan sebagai seni menyembunyikan pesan dalam data lain tanpa mengubah data yang ditumpanginya tersebut sehingga data yang ditumpanginya sebelum dan setelah proses penyembunyian hampir terlihat sama [5].

Steganografi adalah seni dan ilmu berkomunikasi dengan cara menyembunyikan keberadaan komunikasi itu. Berbeda dengan Kriptografi, dimana musuh diperbolehkan untuk mendeteksi, menangkal dan memodifikasi pesan tanpa bisa melanggar keamanan tempat tertentu yang dijamin oleh suatu *cryptosystem*, tujuan dari steganografi adalah untuk menyembunyikan pesan dalam pesan berbahaya lainnya dengan cara yang tidak memungkinkan musuh apapun bahkan untuk mendeteksi bahwa ada pesan kedua. Secara umum, teknik steganografi yang baik harus memiliki visual/ *imperceptibility* statistik yang baik dan *payload* yang cukup [6].

2.3. ALGORITMA *BLOWFISH*

Blowfish merupakan algoritma kriptografi dengan penggunaan kunci pada blok *cipher* simetris (*symmetric block cipher*) yakni kunci yang digunakan pada proses enkripsi sama dengan kunci yang digunakan pada proses dekripsi dengan data masukan dan keluaran berupa blok-blok data berukuran 64 bit. *Blowfish* dirancang oleh Bruce Schneier pada tahun 1993 yang ditujukan untuk mikroposesor besar (32 bit ke atas dengan *cache* data yang besar).

Blowfish dioptimasi untuk aplikasi dimana kunci tidak sering berubah dikarenakan *Blowfish* menggunakan subkunci yang besar. Subkunci ini harus dihitung sebelum proses enkripsi dan dekripsi data. Algoritma ini terdiri dari dua bagian yaitu *key expansion* dan *data encryption. Key expansion* berfungsi merubah kunci yang besarnya dapat mencapai 448 *bit* menjadi beberapa *array* subkunci dengan total 4168 *byte*. *Data encryption* merupakan proses enkripsi yang terdiri dari iterasi beberapa operasi sederhana sebanyak 16 kali. Setiap iterasi terdiri dari permutasi dan substitusi antara bagian kunci dengan data. Seluruh proses menggunakan operasi penambahan dan XOR (*exclusive or*) pada variabel 32 bit. Tambahan operasi lainnya adalah empat penelusuran tabel (*table lookup*) untuk setiap putaran [7].

2.4. *LEAST SIGNIFICANT BIT* (LSB)

Least Significant Bit (LSB) adalah cara paling umum untuk menyembunyikan pesan. LSB dilakukan dengan memodifikasi bit - bit yang termasuk bit LSB pada setiap *byte* warna pada sebuah piksel. Bit - bit LSB ini akan dimodifikasi

dengan menggantikan setiap LSB yang ada dengan bit - bit pesan rahasia yang ingin disembunyikan. Setelah semua bit pesan rahasia menggantikan bit LSB *file* tersebut, maka pesan rahasia telah berhasil disembunyikan. Metode ini memodifikasi nilai yang paling kurang signifikan dari jumlah bit dalam 1 *byte file carrier*. Bit yang memiliki signifikansi paling tinggi adalah numerik yang memiliki nilai tertinggi (misal, 27 = 128), artinya bila terjadi perubahan pada bit ini akan menghasilkan perubahan yang sangat signifikan.

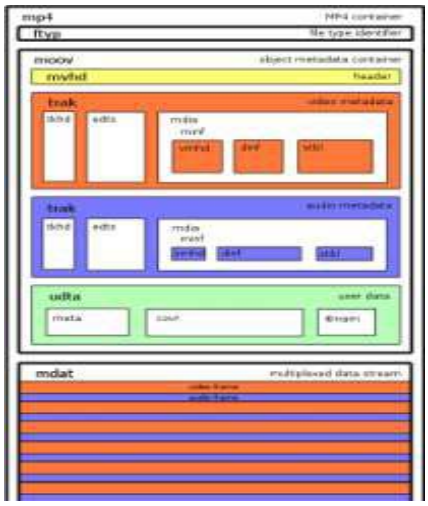
Bit yang memiliki signifikansi paling rendah adalah numerik yang memiliki nilai terendah (misal, 20 = 1), artinya bila terjadi perubahan pada bit ini akan menghasilkan perubahan yang tidak terlalu signifikan. Sebagai contoh, akan dilakukan proses penyembunyian karakter "G" (ASCII 71) pada berkas *carrier* yang berukuran 8 *byte*. *Least Significant Bit* dari *file carrier* ditandai dengan garis bawah.

Berkas *carrier* dalam biner dengan ukuran 8 *byte* :
"10010101 00001101 11001001 10010110 00001111 11001011 10011111 00010000"
Karakter "G" dalam biner dengan ukuran 1 *byte* :
"01000111"
Kedelapan bit ini nantinya akan dimasukan kedalam *Least Significant Bit* dari tiap-tiap *byte* pada *file carrier* seperti berikut ini :
Berkas *carrier* dalam biner dengan ukuran 8 *byte* :
"10010101 00001101 11001001 10010110 00001111 11001011 10011111 00010000"
Karakter "G" dalam biner dengan ukuran 1 *byte* :
" 01000111"
Proses *Least Significant Bit Modification* :
"10010100 00001101 11001000 10010110 00001110 11001011 10011111 00010001"

Pada contoh diatas, hanya sebagian dari *Least Significant Bit file carrier* yang berubah (ditunjukkan dengan karakter miring). Berdasarkan teori yang didapat adalah bahwa kemungkinan terjadinya perubahan bit adalah sekitar 50%, karena peluang perubahannya adalah antara 0 atau 1 dan dengan mengubah *Least Significant Bit* maka ukuran dari *file* pembawa tidak akan berubah sehingga akan sulit untuk terdeteksi [8].

2.5. Mp4

MPEG-4 sub-bagian 14 atau lebih dikenal sebagai MP4 adalah salah satu format berkas pengodean suara dan gambar/video digital yang dikeluarkan oleh sebuah organisasi MPEG. Ekstensi nama berkas jenis MPEG-4 ini banyak menggunakan .mp4, dan merupakan pengembangan dari format QuickTime dari komputer *Apple* Struktur berkas MP4 secara keseluruhan ditunjukkan pada gambar 1.



Gambar 1 Struktur berkas MP4

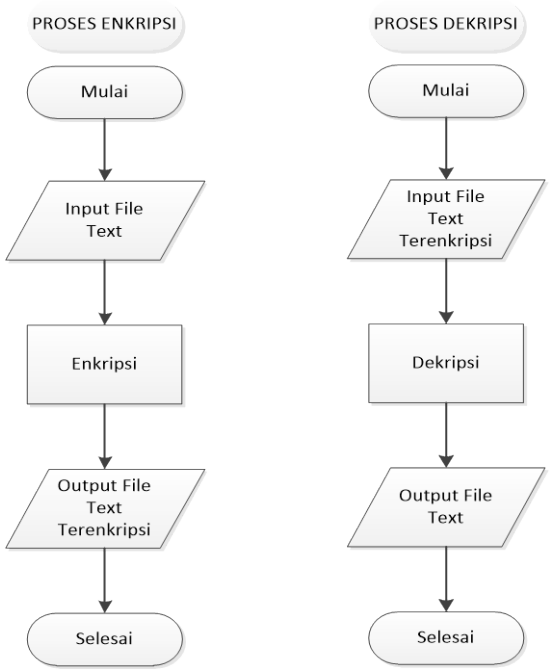
Penggunaan berkas MP4 sebagai wadah untuk steganografi dengan metode *Least Significant Bit* (LSB) akan memanfaatkan posisi data sebagai tempat ditanamkan pesan rahasia, dengan demikian posisi pesan rahasia akan berada dibawah struktur mdat atau *multiplexed data stream* [9].

III. METODE RISET

3.1. Metode Perancangan Sistem

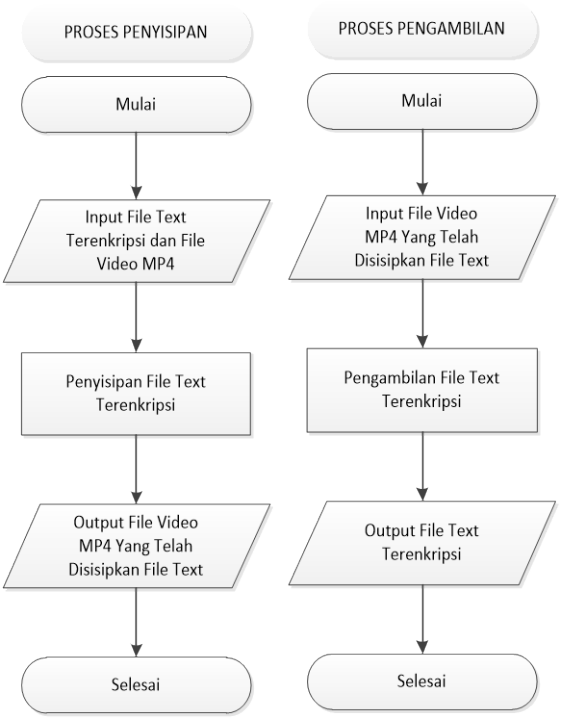
Untuk mendukung penelitian yang dilakukan, penulis melakukan perancangan sistem sesuai dengan kebutuhan, yaitu dimulai dari tahapan *Flowchart* dan Rancangan Aplikasi.

Adapun *flowchart* dalam penerapan aplikasi terbagi menjadi 2 (dua) bagian, yaitu *flowchart* proses kriptografi dan *flowchart* proses steganografi, seperti terlihat pada gambar di bawah ini.



Gambar 2. *Flowchart* Proses Kriptografi

Pada Gambar 2 merupakan gambaran proses kriptografi. Kriptografi merupakan proses pengubahan *file* asli menjadi *file* yang telah dikunci atau diacak. Proses yang terjadi dalam kriptografi dikenal dengan sebutan enkripsi dan dekripsi. Enkripsi merupakan pengubahan dari *plaintext* menjadi *ciphertext*, sedangkan dekripsi merupakan pengubahan dari *ciphertext* menjadi *plaintext*. Proses pengubahan tersebut memerlukan kata kunci baik untuk proses enkripsi maupun proses dekripsi.



Gambar 3. *Flowchart* Proses Steganografi

Pada Gambar 3 merupakan gambaran proses steganografi. Steganografi merupakan proses penyembunyian *file* ke dalam *file* lain, sehingga *file* tersebut tidak terlihat oleh kasat mata. Proses steganografi dibagi menjadi 2 bagian yaitu penyisipan dan pengambilan *file*. Penyisipan *file* merupakan proses dimana dilakukan penyisipan *file* dalam hal ini adalah *file* video MP4 akan disisipkan *file* teks yang telah dienkripsi sebelumnya. Sedangkan pengambilan *file* merupakan proses dimana dilakukan pengambilan *file* yang telah disisipkan ke dalam sebuah video MP4.

IV. HASIL DAN PEMBAHASAN

1. PERHITUNGAN KRIPTOGRAFI

Proses enkripsi algoritma *Blowfish* yang terjadi, yaitu sebagai berikut :

1. Inisialisasi P-*Array* sebanyak 18 buah (P0, P1, ..., P17) masing-masing bernilai 32 bit.
2. Inisialisasi S-*Array* sebanyak 4 buah masing-masing bernilai 32 bit yang memiliki masukan hingga 256, seperti di bawah ini :
   S1.0, S1.1, ... S1.255
   S2.0, S2.1, ... S2.255
   S3.0, S3.1, ... S3.255
   S4.0, S4.1, ... S4.255
3. Memulai proses enkripsi (*plaintext*) dengan X = 64 bit.
4. X dibagi menjadi 2, sehingga terdapat dua bagian yaitu XL (32 bit) dan XR (32 bit).
5. i = 0 merupakan inisial iterasi/perputaran yang dimulai dari 0 (i = i + 1)
6. Memproses fungsi F = XL/4 menjadi a, b, c, d masing-masing 8 bit
7. Memproses F(XL) = (((S0.a + S1.b mod 232) XOR S2, c) + S3,d mod 232)
8. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR
9. Menukar hasil XL dan XR. XL = XR dan XR = XL
10. Melakukan perulangan sebanyak 16 kali.
11. Pada perulangan ke-16, terdapat proses penukaran hasil XL dan XR
12. Setelah proses perulangan selesai pada proses terdapat operasi untuk XR = XR xor P16 dan XL = XL xor P17.
13. Proses terakhir XL dan XR digabungkan kembali sehingga menjadi *ciphertext* 64 bit.
14. Selesai

Proses dekripsi algoritma *Blowfish* yang terjadi, yaitu sebagai berikut :

1. Inisialisasi P-*Array* sebanyak 18 buah (P0, P1, ..., P17) masing-masing bernilai 32 bit.
2. Inisialisasi S-*Array* sebanyak 4 buah masing-masing bernilai 32 bit yang memiliki masukan hingga 256, seperti di bawah ini :
   S1.0, S1.1, ... S1.255
   S2.0, S2.1, ... S2.255
   S3.0, S3.1, ... S3.255
   S4.0, S4.1, ... S4.255
3. Memulai proses dekripsi (*ciphertext*) dengan X = 64 bit
4. X dibagi menjadi 2, sehingga terdapat dua bagian yaitu XL (32 bit) dan XR (32 bit).
5. i = 0 merupakan inisial iterasi/perputaran yang dimulai dari 0 hingga i = 16.
6. j = 17 merupakan inisial pengambilan P-Array dimulai dari P17
7. Memproses fungsi F = XL/4 menjadi a, b, c, d masing-masing 8 bit
8. Memproses F(XL) = (((S1,a + S2,b) XOR S3,c) + S4,d)
9. Selanjutnya lakukan operasi XL = XL xor Pj dan XR = F(XL) xor XR
10. Menukar hasil XL dan XR. XL = XR dan XR = XL
11. Melakukan perulangan sebanyak 16 kali.
12. Setelah perulangan selesai, maka dilanjutkan dengan proses pertukaran hasil XL dan XR
13. Memproses operasi untuk XR = XR xor P1 dan XL = XL xor P0.
14. Proses terakhir XL dan XR digabungkan kembali
15. Menghasilkan *plaintext* 64 bit.
16. Selesai

2. PENGUJIAN PROGRAM

2.1. Pengujian Kriptografi

Hasil pengujian terhadap 5 (lima) *text* diketahui bahwa algoritma *Blowfish* berhasil mengenkripsikan dan mendekripsikan teks yang diinginkan oleh peneliti. Ukuran *file* hasil enkripsi sedikit lebih besar dari ukuran semula tetapi maih kurang dari 1 KB yang dapat kita lihat pada Tabel. 1 dan Tabel. 2 dibawah ini.

Tabel 1. Hasil Pengujian Kriptografi (Proses Enkripsi)

No	Sebelum dienkripsi	Ket. File	Sesudah Dienkropsi	Ket. File
1		Nama File : Text1.txt Size File : 5.06 KB (5.189 bytes) Size On Disk : 8.00 KB (8.192 bytes)		Nama File : Text1_txt.encrypt Size File : 5.07 KB (5.200 bytes) Size On Disk : 8.00 KB (8.192 bytes)
2		Nama File : Text2.txt Size File : 33,0 KB (33.807 bytes) Size On Disk : 36,0 KB (36.864 bytes)		Nama File : Text2_txt.encrypt Size File : 33,0 KB (33.808 bytes) Size On Disk : 36,0 KB (36.864 bytes)
3		Nama File : Text3.txt Size File : 24,6 KB (25.269 bytes) Size On Disk : 28,0 KB (28.672 bytes)		Nama File : Text3_txt.encrypt Size File : 24,6 KB (25.280 bytes) Size On Disk : 28,0 KB (28.672 bytes)
4		Nama File : Text4.txt Size File : 1,97 KB (2.022 bytes) Size On Disk : 4,00 KB (4.096 bytes)		Nama File : Text4_txt.encrypt Size File : 1,98 KB (2.032 bytes) Size On Disk : 4,00 KB (4.096 bytes)
5		Nama File : Text6.txt Size File : 5,97 KB (6.114 bytes) Size On Disk : 8,00 KB (8.192 bytes)		Nama File : Text6_txt.encrypt Size File : 5,98 KB (6.128 bytes) Size On Disk : 8,00 KB (8.192 bytes)

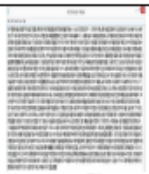
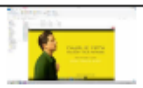
Tabel 2. Hasil Pengujian Kriptografi (Proses Deskripsi)

No	Sebelum dienkripsi	Ket. File	Sesudah Dienkropsi	Ket. File
1		Nama File : Text1_txt.encrypt Size File : 5.07 KB (5.200 bytes) Size On Disk : 8.00 KB (8.192 bytes)		Nama File : Text1.txt Size File : 5.06 KB (5.189 bytes) Size On Disk : 8.00 KB (8.192 bytes)
2		Nama File : Text2_txt.encrypt Size File : 33,0 KB (33.808 bytes) Size On Disk : 36,0 KB (36.864 bytes)		Nama File : Text2.txt Size File : 33,0 KB (33.807 bytes) Size On Disk : 36,0 KB (36.864 bytes)
3		Nama File : Text3_txt.encrypt Size File : 24,6 KB (25.280 bytes) Size On Disk : 28,0 KB (28.672 bytes)		Nama File : Text3.txt Size File : 24,6 KB (25.269 bytes) Size On Disk : 28,0 KB (28.672 bytes)
4		Nama File : Text4_txt.encrypt Size File : 1,98 KB (2.032 bytes) Size On Disk : 4,00 KB (4.096 bytes)		Nama File : Text4.txt Size File : 1,97 KB (2.022 bytes) Size On Disk : 4,00 KB (4.096 bytes)
5		Nama File : Text5_txt.encrypt Size File : 3,68 KB (3.776 bytes) Size On Disk : 4,00 KB (4.096 bytes)		Nama File : Text5.txt Size File : 3,67 KB (3.765 bytes) Size On Disk : 4,00 KB (4.096 bytes)

4.2.2. Pengujian Steganografi

Berdasarkan hasil percobaan pada Tabel. 3 dan Tabel. 4 dapat diketahui bahwa teks yang sudah dienkripsi dapat dimasukkan ke dalam *file* Mp4 dan di ambil kembali tanpa merubah kualitas *file* Mp4 walaupun ukuran *file* sedikit bertambah. Pertambahan ukuran *file* Mp4 dapat kita lihat pada Tabel. 3 dan Tabel. 4 berikut ini:

Tabel 3. Hasil Pengujian Steganografi (Penyisipan *File*)

No	Sebelum Penyisipan File		Sesudah Penyisipan File
	File Video	File Tek Terenkripsi	
1	 Nama File : video1.mp4 Size File : 30,8 MB (32.398.196 bytes) Size On Disk : 30,8 MB (32.399.360 bytes)	 Nama File : Text1_txt.encrypt Size File : 5.07 KB (5.200 bytes) Size On Disk : 8.00 KB (8.192 bytes)	 Nama File : hasil1.mp4 Size File : 30,9 MB (32.403.409 bytes) Size On Disk : 30,9 MB (32.403.456 bytes)
2	 Nama File : video2.mp4 Size File : 18,9 MB (19.825.011 bytes) Size On Disk : 18,9 MB (19.828.736 bytes)	 Nama File : Text2_txt.encrypt Size File : 33,0 KB (33.808 bytes) Size On Disk : 36,0 KB (36.864 bytes)	 Nama File : hasil2.mp4 Size File : 18,9 MB (19.858.833 bytes) Size On Disk : 18,9 MB (19.861.504 bytes)
3	 Nama File : video3.mp4 Size File : 62,1 MB (65.131.864 bytes) Size On Disk : 62,1 MB (65.134.592 bytes)	 Nama File : Text3_txt.encrypt Size File : 24,6 KB (25.280 bytes) Size On Disk : 28,0 KB (28.672 bytes)	 Nama File : hasil3.mp4 Size File : 62,1 MB (65.157.158 bytes) Size On Disk : 62,1 MB (65.159.168 bytes)
4	 Nama File : video4.mp4 Size File : 138 MB (145.562.496 bytes) Size On Disk : 138 MB (145.563.648 bytes)	 Nama File : Text4_txt.encrypt Size File : 1,98 KB (2.032 bytes) Size On Disk : 4,00 KB (4.096 bytes)	 Nama File : hasil4.mp4 Size File : 138 MB (145.564.541 bytes) Size On Disk : 138 MB (145.567.744 bytes)

Tabel 4.Hasil Pengujian Steganografi (Pengambilan *File*)

N0	Sebelum Pengambilan File	Sesudah Pengambilan File
1	 Nama File : hasil1.mp4 Size File : 30,9 MB (32.403.409 bytes) Size On Disk : 30,9 MB (32.403.456 bytes	 Nama File : Text1_txt.encrypt Size File : 5.07 KB (5.200 bytes) Size On Disk : 8.00 KB (8.192 bytes)
2	 Nama File : hasil2.mp4 Size File : 18,9 MB (19.858.833 bytes) Size On Disk : 18,9 MB (19.861.504 bytes)	 na File : Text2_txt.encrypt e File : 33,0 KB (33.808 bytes) Size On Disk : 36,0 KB (36.864 bytes)
3	 Nama File : hasil3.mp4 Size File : 62,1 MB (65.157.158 bytes) Size On Disk : 62,1 MB (65.159.168 bytes)	 Nama File : Text3_txt.encrypt Size File : 24,6 KB (25.280 bytes) Size On Disk : 28,0 KB (28.672 bytes)
4	 Nama File : hasil4.mp4 Size File : 138 MB (145.564.541 bytes) Size On Disk : 138 MB (145.567.744 bytes)	 Nama File : Text4_txt.encrypt Size File : 1,98 KB (2.032 bytes) Size On Disk : 4,00 KB (4.096 bytes)

V. PENUTUP

5.1. KESIMPULAN

Kesimpulan pada penelitian ini yaitu algoritma *Blowfish* dan metode *Least Significant Bit Insertion* dapat diterapkan pada *file* video Mp4 tanpa mengakibatkan perubahan yang signifikan pada *file* Mp4 asalnya.

5.2. SARAN

Kelemahan pada penelitian ini yaitu jika kunci tidak ditetapkan sebelum proses penyisipan dilakukan, maka data tersebut tidak dapat dibuka atau teracak atau rusak, hal ini dapat diperbaiki pada penelitian selanjutnya.

VI. REFERENSI

[1] Ariyus, Doni, 2006, *"Kriptografi Keamanan Data Dan Komunikasi"*, Graha Ilmu: Yogyakarta.

[2] Munir, Rinaldi, 2006, "*Kriptografi*", Informatika: Bandung, 2006.

[3] Cox, I., Miller, M., Bloom, J., & Fridrich, J &."*Digital Watermarking and Steganography 2nd Ed*. Morgan Kaufmann,. MA, 2008

[4] Kipper, G.(n.d), 2004, *"Investigator Guide to Steganograf"*, CRC Press LLC: Florida.

[5] Ariyus, Doni, 2009, ".Keamanan Multimedia", Andi: Yogyakarta.

[6] H. B. Kekre, A. A. "*Increased Capacity of Informationo Hiding In Lsb's Method For Text And Image.* International Journal of Electrical, Computer, and System Engineering, Vol. 2, No.4, p.246-249,2008

[7] Nani, Paskalis Adrianus, 2011, "*Penerapan Enkripsi Algoritma Blowfish Pada Proses Steganografi Metode EOF"*, Universitas Katolik Widya Mandira, Kupang.

[8] Bender, dkk, 1996 "*Techniques For Data Hiding*", IBM Systems Journal.

[9] Azhari,Ahmad Ihsan, 2012, "*Aplikasi Steganografi pada Berkas Video Mp4 dengan Menggunakan Bahasa Pemrograman Java"* Universitas Diponegoro: Semarang.