



Steganografi Video Menggunakan Metode *End Of File* dan Metode *Caesar Chiperext*

Darmawati^{1*}, Muh. Basri²

^{1*}Program Studi Teknik Informatika, Universitas Muhammadiyah Parepare, Indonesia

²program studi Teknik Elektro, Universitas Muhammadiyah Parepare, Indonesia

*Email : darmawati074@email.com

Abstract: the application of technology in the field of data and information security has been widely implemented. Threats to information security can occur when the information sent is not addressed to everyone but only to certain people, especially if the information provided is confidential. The aim of this research is to develop software that can hide confidential information on media. The method used is end of file in the encryption process, and to insert text into the video using the caesar chipertext method, using the php programming language. Based on the test carried out, an application was produced that can be used to hide information using video. The result of this research are changes in the size of the video namely the size of the original video compared to the size of the video after being inserted has increased

Keywords: Steganografi, Video, EOF, Caesar Chiperext;

1. PENDAHULUAN

Steganografi merupakan salah satu cara untuk menyembunyikan suatu pesan atau data rahasia di dalam suatu media penampungnya sehingga orang lain tidak menyadari adanya pesan di dalam media tersebut. Tujuan penggunaan dari steganografi adalah untuk menyamarkan eksistensi data rahasia sehingga sulit di deteksi. Dalam bidang keamanan komputer, steganografi digunakan untuk menyembunyikan data rahasia. Banyak aplikasi teknologi telah dibuat di bidang keamanan data dan informasi. Resiko keamanan informasi mungkin timbul karena pengiriman informasi ke penerima tertentu, bukan ke semua orang, khususnya jika informasi tersebut bersifat pribadi. Ada dua buah proses dalam steganografi yakni proses penyisipan pesan dan proses ekstraksi pesan. Proses penyisipan pesan membutuhkan masukan media penyisipan, Keluaran dari proses penyisipan ini adalah media yang telah berisi pesan.

Saat ini banyak sekali kejahatan yang terjadi di dunia maya, dimana seorang hacker dapat mencuri data sensitif secara diam-diam. Maraknya kasus pencurian data digital ini mulai menyedot perhatian publik, salah satunya para pegiat ilmu forensik digital dan masih banyak lainnya.

Sehingga peneliti tergerak untuk menemukan cara untuk mengamankan data digital dengan efisien dan cepat dari pihak-pihak yang tidak bertanggung jawab, salah satunya dengan menggunakan teknik steganografi. Dengan menggunakan teknik

steganografi isi informasi yang ingin dikirim akan lebih aman karena terlebih dahulu disisipkan ke dalam suatu bentuk *cover object*. *Cover Object* bertujuan untuk pesan yang dikirim tidak terlihat mencurigakan untuk menyimpan pesan rahasia (Septya Maharani, Ismiatul Maula, Zainal Arifin.2016).

Berdasarkan pada penelitian terkait yang meneliti tentang steganografi yaitu, Aplikasi steganografi *pdf to image* dengan metode *spread spectrum* pada tahun 2022. Hasil dari pengujian ini adalah aplikasi yang dapat menyembunyikan informasi dengan gambar sebagai persembunyiaan, perubahan ukuran gambar sesudah disisipkan file pdf dan perubahannya tergantung dari hasil spread spectrum-nya (Nurrahmi Wahyuni., 2022).

Penelitian tersebut selaras dengan penelitian lain yaitu Aplikasi steganografi menggunakan metode LSB (*Leasi Significant Bit*) pada media audio pada tahun 2019. Hasil dari pengujian ini adalah aplikasi yang dapat menyembunyikan informasi ke teks dengan cover audio dengan ukuran file audio, bit rate dan panjang audionya sesuai dengan aslinya (Andi Herul., 2019).

Hal demikian selaras dengan penelitian lain yaitu Aplikasi perpaduan enkripsi algoritma base64 dengan metode *steganografi discrete cosina transform* (DCT) pada tahun 2021. Hasil dari pengujian ini adalah membuat aplikasi enkripsi teks dengan algoritma base64 lalu disisipkan kedalam gambar dengan metode steganografi DCT (Selpi., 2021).

Dengan keterbaruan dari penelitian ini yaitu steganografi video menggunakan metode *end of file* dan metode *caesar chipertext*. Aplikasi yang dibuat berbeda dari penelitian sebelumnya karena menggunakan media video yang menggunakan dua metode sehingga lebih memaksimalkan kinerja aplikasi.

Adapun tujuan penelitian ini adalah merancang sebuah aplikasi yang dapat menyisipkan pesan teks ke dalam sebuah video dengan menggunakan metode *end of file* (EOF) dan metode *caesar chipertext*, Serta mengungkapkan kembali pesan tersebut dari hasil penyisipan tersebut.

2. METODOLOGI PENELITIAN

2.1. Jenis Penelitian

Penelitian ini menggunakan penelitian eksperimental, untuk melakukan percobaan atau eksperiment untuk mengumpulkan data secara langsung dari hasil percobaan yang dilakukan. Penelitian dilakukan di Universitas Muhammadiyah Parepare yang berada di kota parepare, Sulawesi Selatan yang di laksanakan selama dua bulan

2.2. Alat dan Bahan Penelitian

Perangkat Keras	Perangkat Lunak
ACER	Windows 10
Intel(R) Celerom(R) CPU N3050 160Hz	PHP
2 GB	XAMPP

500 GB	Sublime Text
Monitor : 14 inch FHD (1920x1080)	Browser Google Chrome

Bahan yang digunakan dalam penyisipan pesan ini adalah file video .MP4. dengan kapasitas 6 Mb

2.3. Metode End Of File

a. Enkripsi metode end of file

1. Melakukan Penyelipan. Setelah file sudah siap yang dibutuhkan yaitu text tersembunyi yang telah dikonversi ke chipertext dan video sebagai wadahnya, selanjutnya adalah menyelipkan menggunakan metode eof, text yang akan membatasi kedua data tersebut adalah `{eof}{batas}` dimana bentuk hexadesimalnya adalah ditentuka "7B 65 6f 66 7D 62 61 74 61 73 7D" Setelah digabung akan disimpan kedalam file yang akan menjadi representasi dari hasil penyelipan.
2. Membuat video hasil akhir. Proses ini adalah membuat video dengan mengkonversi file hexadesimal proses sebelumnya menjadi bentuk biner. Proses akan menghasilkan video yang dapat dijalankan dan telah diselipkan pesan tersembunyi.

b. Dekripsii Metode End Of File

1. Proses penyelipan pesan.
Setelah melakukan proses enkripsi dan mendapatkan *chipertext* dengan metode *caesar chipertext* dan kemudian hasil chipertext tersebut diubah menjadi bilangan hexadesimal menjadi 4F 52 59 48 untuk disembunyikan kedalam video dengan metode *End Of File*. Kemudian yang akan membatasi antara video dan text tersebut adalah `{eof}{batas}`. Setelah hasil chipertext diubah menjadi hexadesimal selanjutnya pesan akan disembunyikan pada akhir file video dengan nilai 4f 52 59 48. Proses penyimpanan akan menyimpan pesan yang telah disembunyikan pada akhir file video tersebut.
2. Proses membaca pesan
Setelah melakukan 5 putaran output tahap enkripsi selesai dan didapatkan chipertext : 4F 52 59 48 = ORYH. Maka didapatkanlah plainteks dari *chipertext* ORYH adalah LOVE.

2.4 Metode Caesar Chipertext

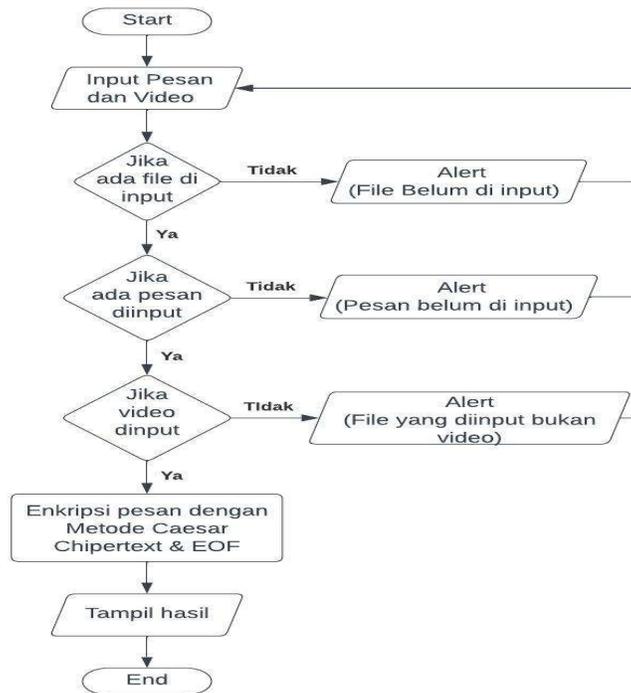
proses Enkripsi suatu huruf P dengan pergeseran K dapat dinyatakan secara matematis sebagai berikut :

$$\text{Enkripsi : } C = E(P) = (P+K) \text{ mod } 26$$

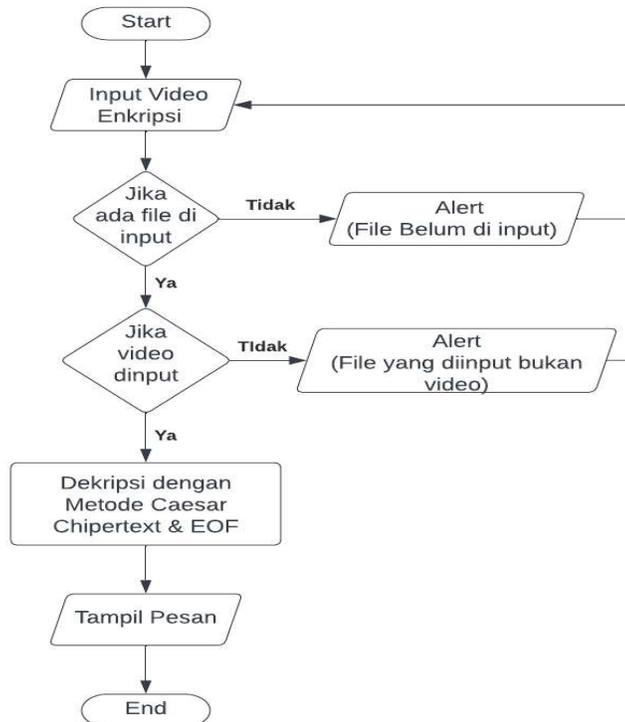
$$\text{Dekripsi : } P = D(C) = (C-K) \text{ mod } 26$$

Dengan C sebagai *chipertext*, P sebagai *plaintext*, K sebagai *secret key*, dan E(P) sebagai enkripsi dan dekripsi,

2.5. Rancangan Sistem



Gambar 1. Proses Enkripsi



Gambar 2. Proses Dekripsi

3. HASIL DAN PEMBAHASAN

3.1. Tampilan Aplikasi

a. Tampilan Halaman Utama

Pada tampilan ini merupakan tampilan awal pada saat kita membuka aplikasi, disini terdapat nama aplikasi *vMessage*, judul aplikasi, mulai, menu enkripsi disini user dapat menginput pesan dan file video yang akan di enkripsi, terdapat juga menu pilihan enkripsi jika ingin melanjutkan proses enkripsi dan mengeluarkan hasil dalam bentuk video.



Gambar 4. Tampilan menu enkripsi

b. Proses Enkripsi

Pada proses enkripsi video dan teks yang ingin disembunyikan di-input dan ditampilkan dalam bentuk hexadesimal agar lebih ringkas dan mudah dibaca. Untuk proses *Caesar Chiptertext* tidak benar-benar seketika bisa dibaca, proses ini menggunakan $key = 3$, artinya kode huruf akan digeser sebanyak 3 kali. Sesuai dengan default dari Caesar karakter yang di enkripsi hanyalah A-Z a-z. Pada proses enkripsi ini adalah proses *End Of File* dimana *Caesar Chiptertext* akan diselipkan pada akhir video, agar mengetahui batas antara video dan teks yang disembunyikan, maka ditentukan "7B 65 6f 66 7D 62 61 74 61 73 7D" sebagai pembatas bentuk teks dari bilangan hex tersebut adalah `{eof}{batas}`.

Enkripsi

Proses enkripsi text ke dalam video

Langkah 1

Pada langkah ini video dan text yang ingin disembunyikan di-input, dan ditampilkan dalam bentuk hexadecimal agar lebih ringkas dan mudah dibaca.

Langkah [1]:

_enkripsi_sesudah_fileHasilVideo.mp4

Masukkan Text Tersembunyi

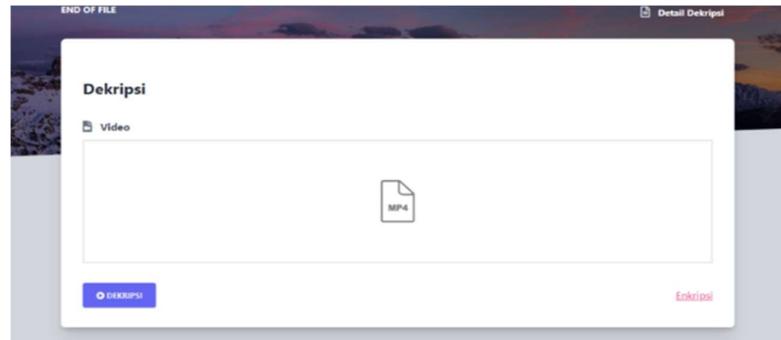
Hasil [1]:

Hex Video:	Raw Text:
00 00 00 18 66 74 79 70 6d 70 51 64 65 6b	Hex Text:
6d 6a 64 67 6b 6b 6d 64 76 6a 6b 76 67 69 6a 6b 67	
6d 6e 76 64 63 6d 71 65 79	

Gambar 6. Proses enkripsi

c. Tampilan Menu Dekripsi

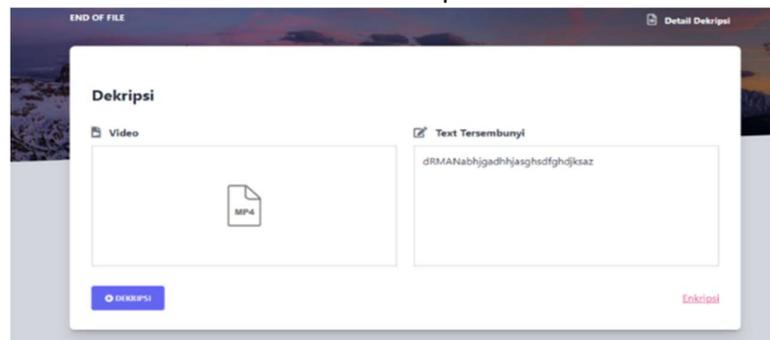
Pada halaman ini user dapat memasukkan video yang berisi pesan atau sudah terenkripsi.



Gambar 10. Tampilan menu dekripsi

d. Hasil Dekripsi

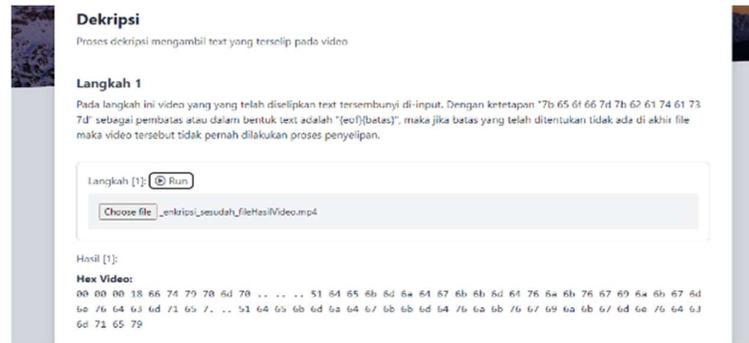
Gambar di bawah menjelaskan setelah user menginput video yang sudah dienkripsi maka pesan tersebut bisa user baca setelah didekripsi.



Gambar 12. Tampilan hasil dekripsi

e. Proses Dekripsi

Proses dekripsi yaitu video yang telah diselipkan text tersembunyi di-input dengan ketetapan "7B 65 6f 66 7D 62 61 74 61 73 7D" sebagai pembatas atau dalam bentuk text adalah `{eof}{batas}`. maka jika batas yang telah ditentukan tidak ada diakhir file maka video tersebut tidak dilakukan proses penyelipan text akan dikeluarkan dari video dengan melakukan pemotongan berdasarkan hal yang telah ditentukan yaitu "7B 65 6f 66 7D 62 61 74 61 73 7D" atau `{eof}{batas}`, text yang telah dipisahkan akan dilakukan decode menggunakan *Caesar Chipertext* dengan $key = 3$ agar menentukan hasil text yang dapat dibaca. langkah untuk membuat kembali video text tersembunyi telah dikeluarkan agar video bersih dari text tersembunyi. Keunggulan dari metode *End Of File* adalah menggunakan penyelinapan text kedalam video tanpa mempengaruhi kualitas visual video.



Gambar 13. Proses dekripsi langkah pertama

3.2. Pengujian Algoritma

Pada tahap pengujian algoritma penulis melakukan pengujian terhadap algoritma yang telah digunakan. Pengujian yang dilakukan untuk mengetahui apakah algoritma yang digunakan hasilnya sama dengan aplikasi yang telah dirancang. Pada pengujian algoritma ini menggunakan *End Of File* dan *Caesar Chipertext*.

Pada dasarnya enkripsi memiliki tiga proses yaitu:

c. Enkripsi

1. Membuat file video

Proses ini memindahkan file video yang mau diselipkan text ke dalam folder aplikasi untuk nantinya diproses sebagai file awal yang belum pernah di proses.

2. Membuat file text yang diselipkan

Proses ini menyimpan text yang ingin disembunyikan, dalam bentuk text yang asli atau tidak pernah dikonversi, sebagai file untuk melihat pesan tersembunyi apa yang akan diselipkan.

3. Membuat file hex untuk video

Proses ini untuk mengkonversi biner video ke bentuk hexadesimal, sebagai representasi dari file vido asli yang belum pernah di proses.

4. Membuat file hex untuk text tersembunyi

Proses ini untuk mengkonversi file text ke bentuk hexadesimal, sebagai representasi dari pesan tersembunyi.

d. Enkripsi metode *caesar chipertext*

1. Membuat file konversi text tersembunyi ke chipertext. Proses ini mengkonversi file text tersembunyi menggunakan metode *caesar chipertext* dengan key yang dimasukkan user, sebagai file untuk melihat hasil konversi chipertext.

2. Membuat *hex* untuk file *chipertext*. Setelah dikonversi menggunakan caesar chipertext dan membuat file chipertextnya, selanjutnya akan dikonversi lagi kedalam bentuk hexadesimal, sebagai file representasi untuk melihat versi hexadesimal dari pesan tersembunyi yang telah dikonversi ke *chipertext*.

Adapun proses dekripsi memiliki tiga proses yaitu:

a. Dekripsi

1. Membuat file video. Proses ini memindahkan video yang diinput ke folder aplikasi untuk nantinya diproses, sebagai file awal yang akan di dekripsi.
 2. Membuat file hexadesimal video. Proses ini adalah proses untuk mengkonversi video yang di input menjadi hexadesimal, sebagai representasi bentuk hexadesimalnya.
- b. Dekripsi
1. Mengeluarkan text tersembunyi. Proses ini adalah proses mengeluarkan text tersembunyi ada dalam video. Isi video dalam bentuk hexadesimal akan di cek apakah terdapat pembatas di dalamnya yaitu `{eof}{batas}` atau dalam bentuk hexadesimalnya "7B 65 6f 66 7D 62 61 74 61 73 7D" jika pembatas itu ada didalam video. Jika pembatas itu ada maka text tersembunyi ada di dalam video, jika tidak ada maka proses akan dihentikan tetapi jika ditemukan program akan mengambil hexadesimal setelah pembatas tersebut dan akan menyimpan sebagi hexadesimal dari text yang diselipkan.
 2. Bentuk text dari hexadesimal hasil potongan. Setelah mengambil potongan hexadesimal maka proses ini akan mengkonversi hexadesimal tersebut kedalam bentuk text agar bisa dibaca, sebagai representasi bentuk karakter dari hexadesimal yang ditemukan.
 3. Dekripsi chipertext. Setelah ditampilkan karakter text yang diselipkan maka akan ditemukan sebuah text unqi. Text unqi ini adalah hasil enkripsi *chipertext* yang saat enkripsi penyelipan dilakukan. Maka untuk dapat melihat text yang sebenarnya maka akan dilakukan dekripsi cipertext denga key yang telah diinput. Key ini harus sama dengan key yang dimasukkan saat enkripsi xhipertext. Untuk default aplikasinya menggunakan key = 3. Setelah dilakukan dekripsi *chipertext* maka akan mendapatkan hasil text sesuai dengan yang diselipkan.
 4. Membuat hexadesimal dari text yang diselipkan. Proses ini mengkonversi text yang telah didekripsi *chipertext* ke hexadesimal dari text yang telah dibuat.
- c. Dekripsi
1. Mengambil Hexadesimal video. Sebelumnya telah dibuat file hexadesimal video yang diinput di dekripsi. File tersebut di proses kembali dan mengambil hexadesimal sebelum pembatas. Pada proses ini yang diambil ada sebelum pembatas sebagai hexadesimal asli dari video tanpa hexadesimal text yang diselipkan.
 2. Proses membuat video. Setelah mengambil hexadesimal video selanjutnya mengkonversinya kedalam bentuk biner untuk membuat file videonya dengan nama filevideo.mp4. Untuk extensi video disesuaikan dengan yang diinput pada proses dekripsi ini.

Tabel 2. Tabel satuan abjad atau alphabet pada *caesar chipertext*

Abjad/Alphabet	Nilai Urut	Abjad/Alphabet	Nilai Urut
A	0	N	13
B	1	O	14
C	2	P	15
D	3	Q	16

E	4	R	17
F	5	S	18
G	6	T	19
H	7	U	20
I	8	V	21
J	9	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25

Bentuk pengujian algoritma *Caesar Chipertext* menggunakan plaintext dilihat pada contoh berikut:

a. Tahapan Enkripsi

1. Input

Langkah proses enkripsi chipertext yaitu:

Plaintext : LOVE

K : 3

$$\begin{aligned} \text{Cp1} &= (\text{Pt1} + k) \text{ modulo } 26 \\ &= (11 + 3) \text{ modulo } 26 \\ &= 14 \text{ modulo } 26 \\ &= 14 = \text{O} \end{aligned}$$

$$\begin{aligned} \text{Cp2} &= (\text{Pt2} + k) \text{ modulo } 26 \\ &= (14 + 3) \text{ modulo } 26 \\ &= 17 \text{ modulo } 26 \\ &= 17 = \text{R} \end{aligned}$$

$$\begin{aligned} \text{Cp3} &= (\text{Pt3} + k) \text{ modulo } 26 \\ &= (21 + 3) \text{ modulo } 26 \\ &= 24 \text{ modulo } 26 = \text{Y} \end{aligned}$$

$$\begin{aligned} \text{Cp4} &= (\text{Pt4} + k) \text{ modulo } 26 \\ &= (4 + 3) \text{ modulo } 26 \\ &= 7 \text{ modulo } 26 = \text{H} \end{aligned}$$

Enkripsi : ORYH

Hexadesimal : 4F 52 59 48 (mencari kode ASCII dari masing-masing huruf)

Deretan binary dijadikan satu baris panjang

"ORYH" : 01001111010100100100101100101001000

2. Proses

Bit : 01001111 01010010 01011001 010010000

No	Biner	Hexadesimal	Char
1	01001111	4F	O
2	01010010	52	R
3	01011001	59	Y
4	01001000	48	H

3. Output

Dari tabel diatas didapat hasil enkripsi ORYH, jadi pesan LOVE dienkripsi menggunakan *Caesar Chipertext* menjadi ORYH.

b. Tahapan Dekripsi

Langkah proses dekripsi chipertext yaitu:

Plaintext : ORYH
 K : 3

$$\begin{aligned}
 \text{Cp1} &= (\text{Pt1} - k) \text{ modulo } 26 & \text{Cp2} &= (\text{Pt2} - k) \text{ modulo } 26 \\
 &= (\text{O} - 3) \text{ modulo } 26 & &= (\text{R} - 3) \text{ modulo } 26 \\
 &= (14 - 3) \text{ modulo } 26 & &= (17-3) \text{ modulo } 26 \\
 &= 11 \text{ modula } 26 & &= 14 \text{ moduula } 26 \\
 &= 11 = \text{L} & &= 14 = \text{O}
 \end{aligned}$$

$$\begin{aligned}
 \text{Cp3} &= (\text{Pt3} - k) \text{ modulo } 26 & \text{Cp4} &= (\text{Pt4} - k) \text{ modulo } 26 \\
 &= (\text{Y} - 3) \text{ modulo } 26 & &= (\text{H} - 3) \text{ modulo } 26 \\
 &= (24 - 3) \text{ modulo } 26 & &= (7-3) \text{ modulo } 26 \\
 &= 21 \text{ modula } 26 & &= 4 \text{ moduula } 26 \\
 &= 21 = \text{V} & &= 4 = \text{E}
 \end{aligned}$$

Untuk proses dekripsi atau decoding tidak jauh berbeda dengan proses endcoding, hasil decoding kemudian di rubah kedalam bentuk desimal berdasarkan tabel ASCII dan dilanjutkan dengan mengubah kebentuk biner 8bit.

Decoding = LOVE
 Hexadesimal = 4C 4F 56 45
 Biner 8 Bit = 01001100 01001111 01010110 01000101

Kemudian hasil binary 8 bit ke kode ASCII, maka hasil text akan balik ke semula yaitu LOVE.

KESIMPULAN

Berdasarkan implementasi dan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa Aplikasi ini berjalan pada komputer yang merupakan Steganografi dengan menyisipkan pesan teks ke dalam sebuah video untuk membantu dan memudahkan pengguna dalam mengirim pesan rahasia tanpa rasa khawatir. Dengan lebih spesifik dimana metode *End Of File* dan *Chipertext* untuk menyisipkan teks, membuat aplikasi steganografi video. Browser web digunakan untuk tampilan, dan Visual Studio Code digunakan untuk desain aplikasi selama konstruksi sistem. Pada pengujian ini apa bila video tidak lebih dari 6 Mb maka pengujian itu bisa di lakukan tetapi jika video melebihi dari 6 Mb maka pengujian tersebut tidak dapat di lakukan. File video terenkripsi akan disimpan di penyimpanan yang tersedia setelah diunduh. Sementara itu, sistem terus diuji untuk mencapai hasil yang diinginkan. Aplikasi ini dibuat dengan tujuan untuk dapat membantu pengguna/ user untuk mengirim sebuah pesan teks bersifat rahasia, agar pesan dapat sampai ketangan penerima tanpa menimbulkan kecurigaan pada pihak lain.

REFERENSI

- Andi. H. (2019). "Aplikasi Steganografi Menggunakan Metode LSB (*Least Significant Bit*)". Master of Computer Science.
- Ariska, A., & Wahyuddin, W. (2022). Penerapan Kriptografi Menggunakan Algoritma Des (Data Encryption Standard). *Jurnal Sintaks Logika*, 2(2), 9-19.
- Ayu, A. N. S. (2023). Aplikasi Pembaca Nilai Resistor Berbasis Android. *Jurnal Sintaks Logika*, 3(1), 17-22.
- Awaluddin. (2022). Simbol simbol pada Flowchart dan penjelasannya. *Simbol simbol pada Flowchart dan penjelasannya*.
- Desi Darwis, Kisworo. *Jurnal Sistem Informasi dan Telematika (Telekomunikasi, Multimedia dan Informatika) 8 (2), 2017*
- Ferdy, F., & Wahyuddin, W. (2024). Aplikasi Game Edukasi Mitigasi Bencana Alam (Gempa Bumi Dan Tsunami) Menggunakan Metode Waterfall Berbasis Android. *Jurnal Sintaks Logika*, 4(1), 1-6.
- Fuad, Hilmi, Sutarman, Yahya. 2018. *Perancangan Sistem Informasi Costumer Relationship Management Pelayanan Berbasis Web Di PT Sahabat Kreasi Muda*. JURNAL SISFOTEK GLOBAL. Vol. 8 No. 1. ISSN : 2088-1762
- MADCOMS. 2016. Manajemen sistem jaringan komputer. Andi
- Maimunah. (2017). Pengertian dan Penjelasan PHP (Hypertext Preprocessor). *Pengertian dan Penjelasan PHP (Hypertext Preprocessor)*.
- Mundzir, M.F. (2018). *Buku Sakti Pemrograman Web Seri PHP*. START UP.
- Nurrahmi Wahyuni. (2022). Aplikasi Steganografi Pdf To Image Dengan Metode *Speed Spectrum*.
- Rerung, R.R. (2018). Pemrograman web dasar. Yogyakarta : Deeppublish
- Samsuriah, Ida. (2023). *Penerapan Kriptografi Caesar Chiper Pada Fitur Pesan Teks*. Nusantara Hasana Journal. Vol. 2 No. 9 E.ISSN: 2798-1428
- Selpi. (2021). Aplikasi Perpaduan Enkripsi Algoritma *Based64* Dengan Metode Steganografi *Discrete Cosina Transform* (DCT)
- Syamsiah. *Jurnal Perancangan Flowchart Dan Pseudocode Pembelajaran Mengenal Angka Dengan Animasi Untuk Anak Paud Rambutan*. (Program Studi Informatika, Universitas Indraprasta PGRI). STRING Vol.4 No 1 Agustus 2019 p-ISSN: 2527-9661 e-ISSN: 2549-2837
- Septya Maharani, Ismiatul Maula, Zainal Arifin. 3 Oktober 2016. Steganografi

- video menggunakan metode *End Of File (EOF)*, *Universitas Mulawarman, Kalimantan Timur*
- Wahyuddin, W., & Hasim, A. (2023). Aplikasi Ekstraksi Data Kartu Vaksin Berbasis Web Menggunakan Metode Ocr. *Jurnal Sintaks Logika*, 3(2), 53-57.
- Wahyuddin, W., & Saputra, A. (2021). Aplikasi schedule pengerjaan proyek online dinas PU Kab. Sidrap. *Jurnal Sintaks Logika*, 1(2), 54-61.
- Wahyuddin, W., & As, K. (2022). Pengembangan Aplikasi Risalah Tuntunan Shalat Secara Lengkap Berbasis Android. *Jurnal Sintaks Logika*, 2(1), 248-256.
- Wahyuddin, W., & Wafiah, A. (2022). Aplikasi Pemesanan Menu Pada Warkop Shearlock Berbasis Abdroid. *Jurnal Sintaks Logika*, 2(3), 11-16.
- Wahyuddin, W., Alam, S., & Said, I. R. (2021). E-COMMERCE BUMBU MASAKAN KELOMPOK TANI KWT (KELOMPOK WANITA TANI) SETIA DESA PAKKODI KAB. ENREKANG. *Jurnal Sintaks Logika*, 1(3), 209-214.
- Wahyuddin, W., & Hasnawati, H. (2023). Aplikasi Media Pembelajaran Pengenalan Pancasila Menggunakan Augmented Reality. *Jurnal Sintaks Logika*, 3(3), 8-15.