

BAB I

PENDAHULUAN

A. Latar Belakang

Banyak aplikasi teknologi telah dibuat di bidang keamanan data dan informasi. Risiko keamanan informasi mungkin timbul karena pengiriman informasi ke penerima tertentu, bukan ke semua orang, khususnya jika informasi tersebut bersifat pribadi. Saat ini banyak sekali kejahatan yang terjadi di dunia maya, dimana seorang hacker dapat mencuri data sensitif secara diam-diam. Pemilik informasi pribadi prihatin dengan hal ini.

Menjamin kerahasiaan dan keamanan data dan informasi sangat penting ketika berkomunikasi melalui internet. Dengan kemajuan teknologi saat ini, ada banyak cara untuk mencegah data penting. Harus ada mekanisme yang menjamin kerahasiaan informasi yang akan dikirimkan kepada pihak yang berkepentingan karena jika tidak, pihak yang tidak berkepentingan dapat melihat, membaca, atau mengubah isi asli informasi tersebut. Dengan demikian, teknik penyembunyian data dapat digunakan untuk melindungi informasi yang perlu dikirim. Steganografi merupakan salah satu metode penyembunyian informasi.

B. Rumusan Masalah

Peneliti dapat merumuskan masalah untuk pembahasan berdasarkan latar belakang informasi yang telah diberikan diatas yaitu “bagaimana membangun

sebuah aplikasi yang dapat menyembunyikan pesan pada video dengan menggunakan metode End Of File dan metode Ceasar Chipertext?”

C. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan perangkat lunak yang menggunakan teknik Caesar Chipertext dan End Of File untuk menyelubungi informasi data dengan pesan tersembunyi.

D. Batasan Masalah

Adapun batasan masalah dari penelitian yang dilakukan adalah sebagai berikut :

1. Format video yang digunakan adalah *MP4*
2. Size video maksimal berukuran 6 MB
3. Metode steganografi yang digunakan yaitu *End Of File (EOF)* dan Metode *Caesar Chipertext*
4. Size video ketika disisipkan pesan akan bertambah.

E. Manfaat Penelitian

Manfaat dari penelitian ini sangat relevan dengan tujuan dan permasalahan yang dibahas. Berdasarkan latar belakang dan rumusan masalah, berikut adalah manfaat dari penelitian ini:

1. Penelitian ini dapat menghasilkan perangkat lunak yang memungkinkan untuk menyembunyikan pesan dalam sebuah video.
2. Pengembangan perangkat lunak ini dapat membantu dalam meningkatkan keamanan pesan rahasia. Informasi yang disembunyikan dalam video akan sulit

diakses oleh pihak yang tidak berkepentingan atau yang berusaha untuk mencari tahu.

3. Hasil penelitian ini dapat dijadikan sebagai aplikasi alternatif dalam penyembunyian pesan yang aman pada file video.

F. SISTEMATIKA PENULISAN

Struktur penulisan tugas akhir ini adalah sebagai berikut :

1. BAB I : PENDAHULUAN

berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan masalah, manfaat penelitian, sistematika penulisan dari pembuatan tugas akhir ini.

2. BAB II : TINJAUAN PUSTAKA

Pada bab ini berisi teori-teori yang mendukung dalam pembahasan penyusunan tugas akhir ini.

3. BAB III : METODE PENELITIAN

Bab ini menjelaskan secara detail cara penelitian atau perancangan yang meliputi : jenis penelitian, lokasi dan waktu, alat dan bahan, rancangan sistem, rancangan sistem, teknik pengumpulan data, teknik analisis data, dan diagram alir.

4. BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini membahas tentang hasil penelitian dan pembahasannya.

5. BAB V : PENUTUP

Pada bab penutup ini mencakup kesimpulan saran keduanya dibahas terpisah.

6. DAFTAR PUSTAKA

7. LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Perancangan

Perancangan adalah proses menentukan langkah-langkah dan data yang diperlukan untuk membangun suatu sistem baru. Memberikan gambaran desain yang lengkap kepada pemrogram untuk membantu mereka membuat aplikasi adalah keuntungan dari tahap perancangan sistem ini. Pada tahap ini, basis data, aplikasi, perangkat keras, dan perangkat lunak semuanya harus dirancang dengan mempertimbangkan elemen-elemen yang terdiri dari sistem komputerisasi. Tahapan yang dilakukan dalam perancangan sistem adalah sebagai berikut:

a. Perancangan output

Laporan yang dihasilkan harus memberikan kemudahan bagi setiap elemen manusia yang memerlukannya, sehingga desain keluaran tidak bisa diabaikan.

b. Perancangan input

Tujuan dari desain input adalah untuk memastikan bahwa entri data akan diterima dan dipahami oleh pengguna, mencapai akurasi yang tinggi, dan mengurangi biaya entri data.

c. Perancangan proses sistem

Tujuan dari perancangan proses sistem adalah untuk memastikan bahwa proses data berfungsi dengan baik sehingga dapat memantau proses sistem dan menghasilkan informasi yang akurat.

d. Perancangan database

Kumpulan data yang terintegrasi dengan koneksi antar data disebut database sistem.

e. Tahapan perancangan control

Tujuan perancangan adalah agar sistem dapat diandalkan dalam mencegah kesalahan, kerusakan, dan kegagalan proses sistem setelah digunakan.

2. Analisis

Menganalisis suatu objek melibatkan pemecahannya menjadi bagian-bagian penyusunnya dan menatanya kembali sehingga dapat diperiksa lebih dekat guna mengamati bagaimana objek tersebut berfungsi. Akurasi diperlukan untuk analisis, sebuah langkah penting dalam proses peninjauan data, untuk mencapai tujuan analisis. Adapun tujuan analisis dapat dibagi empat yaitu:

- a. Tujuannya adalah untuk mengidentifikasi beberapa titik data dari populasi tertentu guna menarik kesimpulan untuk pengambilan kebijakan dan strategi pemecahan masalah.
- b. Tetapkan tujuan yang jelas. Tujuannya agar data dapat mengungkap pemahaman yang lebih detail dan komprehensif.

- c. Analisis lebih lanjut diperlukan karena tujuan menghubungkan beberapa titik data dari lingkungan berbeda adalah untuk membuat kesimpulan dan memperoleh pemahaman yang lebih menyeluruh tentang volume data yang dikumpulkan dari berbagai sumber.
- d. Dengan memilih pendekatan yang berbeda, tujuannya adalah untuk memecahkan masalah sesuai dengan kebutuhan dan keadaan.

3. Steganografi

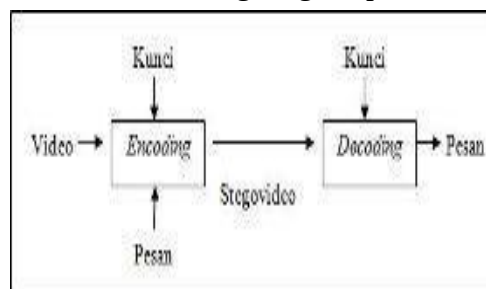
Steganografi adalah praktik atau ilmu dalam seni menyembunyikan pesan rahasia dalam media atau wadah sehingga pesan tersebut tidak terlihat oleh mata telanjang. Ini adalah cara untuk menyembunyikan informasi sensitif atau rahasia dalam data yang tampaknya biasa. Dalam konteks steganografi digital, media digital seperti gambar, suara, teks, dan video digunakan sebagai wadah untuk menyembunyikan data rahasia. Teknik steganografi mengizinkan penanam pesan untuk menyembunyikan pesan rahasia dengan cara yang sulit dideteksi oleh pihak ketiga yang tidak memiliki pengetahuan tentang keberadaan pesan tersebut. Hal ini berbeda dari kriptografi, di mana pesan dienkripsi untuk menjaga kerahasiaannya, sementara steganografi bertujuan untuk menyembunyikan pesan secara fisik dalam wadahnya.

Steganografi memiliki berbagai aplikasi, termasuk keamanan informasi, perlindungan privasi, dan deteksi keaslian data. Hal ini dapat digunakan dalam berbagai konteks, mulai dari komunikasi militer hingga perlindungan data pribadi. Dalam era digital, teknik steganografi telah berkembang dengan cara yang signifikan dan menjadi relevan dalam dunia yang semakin terhubung

secara digital. Dalam bidang keamanan komputer, steganografi digunakan untuk menyembunyikan data rahasia, enkripsi tidak dapat dilakukan atau bersamaan dengan dekripsi. Walaupun enkripsi berhasil dipecahkan (*decipher*), pesan atau data rahasia tetap tidak terlihat. Pada *kriptografi*, pesan disembunyikan dengan “diacak” sehingga pada kasus-kasus tertentu dapat dengan mudah mengundang kecurigaan, sedangkan pada steganografi pesan “disamarkan” dalam bentuk yang relatif lebih “aman” sehingga tidak terjadi kecurigaan itu.

Steganografi selalu memiliki dua proses, yaitu *encoding* dan *decoding*. *Encoding* merupakan proses penyisipan pesan kedalam sebuah media penampung (*cover*) sedangkan *Decoding* adalah proses ekstraksi pesan dari media penampung (*cover*).

Gambar 2. 1 Sistem steganografi pada video



Dalam proses steganografi terdapat kriteria yang harus dipenuhi, kriterianya adalah sebagai berikut:

- a. *Imperceptibility*, keberadaan pesan rahasia tidak dapat dipersepsi oleh indrawi. Misalnya, jika *covertext* berupa citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka indra telinga dapat mendeteksi perubahan pada audio *stegotext*-nya.

- b. *Idelity*, mutu *stegomedium* tidak berubah banyak akibat penyisipan. Perubahan tersebut tidak dapat dipersepsi oleh indrawi. Misalnya, jika *coverttext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *coverttext-nya*. Jika *coverttext* berupa audio, maka audio *stegotext* tidak rusak dan indra telinga tidak dapat mendeteksi perubahan tersebut.
- c. *Recovery* ,Pesan dalam steganografi harus selalu dapat diungkapkan kembali, karena tujuan utamanya adalah menyembunyikan data. Oleh karena itu, setiap saat, pesan rahasia yang telah disisipkan dalam stegoteks harus dapat diambil kembali agar bisa digunakan lebih lanjut.

4. Video Digital

Video digital adalah format video yang menggunakan teknologi digital untuk merekam, menyimpan, dan memproses gambar bergerak. Sebagai lawan dari format analog, video digital menggunakan kode biner (0 dan 1) untuk merepresentasikan informasi gambar dan suara. Teknologi video digital telah mengubah cara kita merekam, menonton, dan berbagi video, dan ini telah menjadi standar dalam industri media.

Sebenarnya, video digital terdiri atas serangkaian gambar digital yang ditampilkan dengan kecepatan konstan. Setiap frame adalah gambar digital yang terdiri dari *raster pixel*, dan satuan ukuran untuk menghitung jumlah frame rata-rata yang ditampilkan disebut *frame per second* (FPS). Adapun beberapa contoh dari format *video digital* yang sering dijumpai antara lain :

a. *AVI (Audio Video Interleave)*

AVI, yang merupakan singkatan dari *Audio Video Interleave*, adalah sebuah format file multimedia yang digunakan untuk menyimpan audio dan video dalam satu file. Format ini dikembangkan oleh *Microsoft* dan diperkenalkan pada tahun 1992 sebagai bagian dari teknologi *Video for Windows*. AVI adalah salah satu format multimedia yang paling umum digunakan dan didukung oleh banyak perangkat lunak pemutar video.

Salah satu kekurangan penggunaan AVI untuk *playback* adalah bahwa *Macintosh*, SGI, dan Sun harus mengubah file ke format lain jika mereka ingin memainkannya. Format AVI memang masih kurang canggih, berbasis trek, dan kemampuan untuk sinkronisasi dengan *Quick Time* kurang baik, dan *codec* untuk *Quick Time* pada Windows lebih berkembang daripada *codec* untuk AVI.

b. *MPEG (Motion Picture Experts Group)*

MPEG, yang merupakan singkatan dari *Motion Picture Experts Group*, adalah sebuah kelompok kerja internasional yang bertanggung jawab untuk mengembangkan standar kompresi video dan audio yang digunakan dalam berbagai aplikasi multimedia. *MPEG* didirikan pada tahun 1988 oleh ISO (Organisasi Internasional untuk Standardisasi) dan IEC (Komisi Elektroteknik Internasional) dengan tujuan untuk menciptakan standar yang memungkinkan penyimpanan dan transmisi data multimedia yang efisien tanpa mengorbankan kualitas. *MPEG* terdiri dari beberapa bagian :

- a) *Synchronization and multiplexing of video and audio*
- b) *Compression codec for non-interlaced video signals*
- c) *Compression codec for perceptual coding of audio signals*

MP1 or MPEG-1 part 3 Layer (MPEG-1 Audio Layer 1)

MP2 or MPEG-1 part 3 Layer (MPEG-1 Audio Layer 2)

MP3 or MPEG-1 part 3 Layer (MPEG-1 Audio Layer 3)

- d) *Procedure for testing conformance*
- e) *Reference for testing*
- f) *Reference software*

MPEG-1: MPEG-1 umumnya memiliki resolusi video 352 x 240 (disebut juga sebagai resolusi CIF - *Common Intermediate Format*) dan mendukung video dengan pemindaian progresif. Ini adalah format yang digunakan untuk Video CD (VCD) dan format multimedia serupa. MPEG-1 umumnya digunakan untuk konten dengan resolusi standar.

c. *MKV*

MKV adalah salah satu format video yang mungkin sering dijumpai di internet. *MKV* merupakan alternatif format video selain beberapa format video digital seperti *AVI*, *MPEG*, *3GP*, *RMVB* dimana masing-masing memiliki sifat dan kualitas yang berlainan. Format *MKV* (Matroska Video) sering digunakan untuk video dengan kualitas tinggi, seperti video definisi tinggi (HD) dan video dengan banyak aliran audio dan subtitle. File video *MKV* memiliki beberapa bagian yang memungkinkan fleksibilitas dalam

penyimpanan dan pengaturan kontennya. Beberapa bagian utama dari file MKV mencakup:

1. Video Stream
2. Audio stream
3. Subtitle stream

Bagian-bagian ini tidak terhubung satu sama lain, tetapi akan digabungkan menggunakan codec MKV sehingga video digital dalam format MKV dapat dibaca dan dijalankan oleh perangkat lunak pemutar multimedia.

5. Algoritma

Sebuah program komputer yang menggunakan serangkaian instruksi yang terorganisir dan terbatas untuk memecahkan masalah komputasi tertentu disebut algoritma.

Fakta bahwa algoritma tidak terbatas pada bahasa pemrograman tertentu menyiratkan bahwa algoritma yang dikembangkan dapat digunakan dengan bahasa pemrograman apa pun. Tidak ada pedoman pasti untuk menulis algoritma, namun tujuan setiap langkah harus dipahami. Namun pada dasarnya algoritma dibagi menjadi beberapa bagian macam berdasarkan format penulisannya yaitu :

a. Deskriptif

Algoritma deskriptif ditulis dalam kalimat menggunakan bahasa umum manusia, seperti bahasa Indonesia atau bahasa lain selain bahasa

Inggris. Langkah-langkah algoritma masing-masing dijelaskan dalam satu kalimat atau lebih.

b. Flowchart (Diagram Alir)

Algoritma dapat ditulis tidak hanya dalam bentuk teks tetapi juga dalam bentuk diagram, dengan panah yang berfungsi sebagai penanda langkah-langkah dalam urutan algoritma. Algoritma simbolik digunakan dalam diagram alur.

6. Metode End Of File (EOF)

Salah satu teknik steganografi adalah metode EOF. Data dimasukkan di akhir file menggunakan metode ini. Oleh karena itu, hal ini tidak akan berdampak pada kualitas data asli tempat pesan akan disisipkan. Namun setelah pesan rahasia disisipkan, ukuran file akan bertambah. Karena ukuran file yang disisipkan akan memperbesar ukuran file yang dihasilkan. *End Of File* menunjukkan awal pesan dengan karakter yang berbeda. Teknik ini memanfaatkan ketidakpekaan indera manusia untuk menciptakan ilusi bahwa pesan disisipkan tanpa membuat perbedaan yang jelas antara sebelum dan sesudahnya.

Pesan dalam steganografi seringkali diubah menjadi nilai desimal berdasarkan tabel ASCII (*American Standard Code for Information Interchange*). Kode ASCII adalah sistem representasi numerik karakter yang digunakan dalam komputer, dan itu mencakup karakter huruf a-z, A-Z, 0-9, dan berbagai simbol standar pada keyboard..

Tabel 2.1. Kode ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Sebagai contoh, pesan bertuliskan "aku" dimasukkan ke dalam gambar grayscale 6x6 pixel. Untuk menandai akhir pesan, karakter yang jarang digunakan, seperti karakter #, digunakan. Pesan yang dimaksud adalah "#aku".

Berikut ini adalah kode ASCII pesan :

97 107 117 35

Misalkan sebuah citra grayscale dengan kode warna :

196 10 97 182 101 40

67	200	100	50	90	50
25	150	45	200	75	28
176	56	77	100	25	200
101	34	250	40	100	60
44	66	99	125	190	200

Nilai desimal pesan berdasarkan tabel ASCII disisipkan diakhir citra, sehingga citra menjadi :

196	10	97	182	101	40
67	200	100	50	90	50
25	150	45	200	75	28
176	56	77	100	25	200
101	34	250	40	100	60
44	66	99	125	190	200
97	107	117	35		

Isi awal file yang disisipkan tetap tidak berubah ketika menggunakan teknik *End Of File*. Misalnya, isi dokumen tetap tidak berubah ketika pengguna memasukkan pesan ke dalamnya. Salah satu keunggulan pendekatan EOF dibandingkan teknik steganografi lainnya adalah ini. Integrasi data dari file yang disisipkan tetap terjaga karena pesan disisipkan di akhir file, sehingga tidak tumpang tindih dengan isinya.

Adapun langkah-langkah encoding menggunakan metode *End Of File* :

- a. Proses dimulai dengan penginputan file video sebagai wadah untuk menyelipkan teks.

- b. Memasukkan pesan teks yang akan diselipkan.
- c. Untuk file video diubah menjadi data biner.
- d. Untuk pesan teks di enkripsi dan diubah ke dalam bentuk biner.
- e. Setelah mendapatkan data biner untuk file video dan data biner untuk pesan teks. Selanjutnya melakukan proses penyelipan di akhir biner file video (EOF).
- f. Proses penyelipan dilakukan dengan memberikan pembatas dengan format <biner video> <biner pembatas> <biner pesan>.
- g. Pembatas diperlukan untuk mengetahui pemisah antara biner dari video dan biner dari teks yang diselipkan.
- h. Karena pesan yang diselipkan diakhir video maka, data yang diselipkan tidak memiliki batasan.
- i. Setelah proses penyelipan akan dibuat gambar yang baru. Gambar baru ini memiliki pesan yang telah diselipkan.

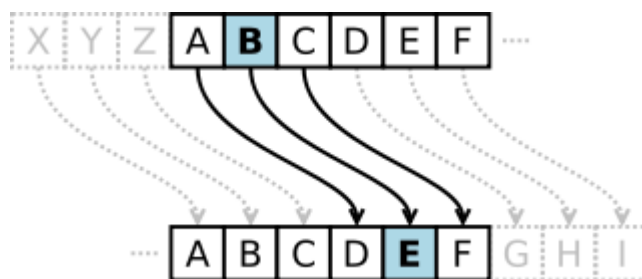
Adapun langkah-langkah proses decoding yang telah disisipkan pesan dengan metode *End Of File* adalah sebagai berikut :

- a. Proses dimulai dengan penginputan file video yang berisi pesan didalamnya.
- b. Ubah file video menjadi biner
- c. Proses pengeluaran pesan dilakukan dengan mendeteksi biner pembatas
- d. Jika tidak berhasil mendeteksi biner pembatas maka file tidak memiliki pesan tersembunyi.

- e. Jika berhasil mendeteksi biner pembatas maka ambil semua data biner setelah file pembatas. Pada proses ini menunjukkan bahwa pesan yang dapat diselipkan tidak memiliki batasan panjang ukuran, selama biner pesannya terletak setelah biner pembatas.
- f. Data biner yang di dapat kemudian di decode
- g. Kemudian biner diubah kedalam bentuk teks menggunakan ASCII
- h. Dengan proses tersebut maka pesan yang diselipkan dapat ditampilkan.

7. Metode Caesar Chipertext

Caesar cipher adalah sejenis sandi substitusi di mana setiap karakter dalam teks biasa diubah menjadi tepat satu karakter dalam teks sandi untuk membuat sandi. *Caesar code*, *Caesar shift*, *shift cypher*, dan *shift cypher* adalah nama lain dari *Caesar cypher*. Kata sandi ini merupakan bagian dari kode substitusi yang menggantikan huruf yang berbeda dari angka tertentu dalam alfabet untuk setiap huruf dalam pesan asli (*plaintext*). Analogi *Caesar Chiper* dapat ditunjukkan pada gambar berikut :



Gambar 2.2 Analogi Caesar Chiper

Proses enkripsi pada *Caesar Chiper* di representasikan menggunakan operator aritmatika *modulo* 26 setelah sebelumnya setiap huruf ditransformasikan ke dalam angka yaitu : A = 0, B =1, ... Z =25. Maka *Caesar*

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

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

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

Dengan C sebagai *chipertext*, P sebagai *plaintext*, K sebagai *secret key*, dan $E(P)$ sebagai enkripsi dan dekripsi, *Caesar Chiper's* kelemahan dapat dipecahkan dengan serangan kekuatan brute, suatu jenis serangan yang mencoba berbagai cara untuk menemukan kunci. Karena jumlah kunci yang sangat kecil (hanya 26 kunci), Anda juga dapat menggunakan pencarian kunci lengkap. Proses dekripsi menggunakan persamaan 1 dibawah ini :

$$C_p = (P_t + k) \bmod 26 \dots\dots\dots (1)$$

Dimana 26 adalah jumlah *alphabet*, persamaan 1 digunakan pada enkripsi, proses dekripsi menggunakan persamaan 2 dibawah ini :

$$P_t = (C_p - k) \bmod 26 \dots\dots\dots (2)$$

Berikut satuan dari abjad atau *alphabet* pada *Caesar Chiper* sebagai berikut:

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

Contoh *Caesar Chiper* :

a. Tahap Enkripsi

Suatu tahap membuat pergantian sebuah sandi dan dapat dipahami (*plaintext*) menjadi sebuah sandi yang tidak dapat dipahami (*chipertext*). Misalkan, diketahui *plaintext* sebagai berikut :

Plaintext : DARMA

K : 3

Kemudian lakukan perhitungan *chipertext* $C_p = (P_t + k) \text{ modulo } 26$ dan cek pada tabel 1 *alphabet* dari nilai *chipertext* yang dihasilkan.

$$\begin{aligned} C_{p1} &= (P_{t1} + k) \text{ modulo } 26 & C_{p2} &= (P_{t2} + k) \text{ modulo } 26 \\ &= (3 + 3) \text{ modulo } 26 & &= (0 + 3) \text{ modulo } 26 \\ &= 6 \text{ modulo } 26 & &= 3 \text{ modulo } 26 \\ &= 6 = G & &= 3 = D \end{aligned}$$

$$\begin{aligned} C_{p3} &= (P_{t3} + k) \text{ modulo } 26 & C_{p4} &= (P_{t4} + k) \text{ modulo } 26 \\ &= (17 + 3) \text{ modulo } 26 & &= (12 + 3) \text{ modulo } 26 \\ &= 20 \text{ modulo } 26 & &= 15 \text{ modulo } 26 \\ &= 20 = U & &= 15 = P \end{aligned}$$

$$\begin{aligned} C_{p5} &= (P_{t5} + k) \text{ modulo } 26 \\ &= (0 + 3) \text{ modulo } 26 \\ &= 3 \text{ modulo } 26 \\ &= 3 = D \end{aligned}$$

Hasil Enkripsi adalah “GDUPD”

b. Tahap Dekripsi

Berkebalikan pada tahap enkripsi yaitu untuk menggantikan sandi dari yang tidak bisa dipahami (*plaintext*). Contoh kasus, jika diberikan *chipertext* sebagai berikut :

Chipertext : GDUPD

K : 3

$$\begin{aligned}
 \text{Cp1} &= (\text{Pt1} - k) \text{ modulo } 26 & \text{Cp2} &= (\text{Pt2} - k) \text{ modulo } 26 \\
 &= (G - 3) \text{ modulo } 26 & &= (D - 3) \text{ modulo } 26 \\
 &= (6 - 3) \text{ modulo } 26 & &= (3 - 3) \text{ modulo } 26 \\
 &= 3 \text{ modulo } 26 & &= 0 \text{ modulo } 26 \\
 &= 0 = D & &= 0 = A
 \end{aligned}$$

$$\begin{aligned}
 \text{Cp3} &= (\text{Pt3} - k) \text{ modulo } 26 & \text{Cp4} &= (\text{Pt4} - k) \text{ modulo } 26 \\
 &= (U - 3) \text{ modulo } 26 & &= (P - 3) \text{ modulo } 26 \\
 &= (20 - 3) \text{ modulo } 26 & &= (15 - 3) \text{ modulo } 26 \\
 &= 17 \text{ modulo } 26 & &= 12 \text{ modulo } 26 \\
 &= 17 = R & &= 12 = M
 \end{aligned}$$

$$\begin{aligned}
 \text{Cp5} &= (\text{Pt5} - k) \text{ modulo } 26 \\
 &= (D - 3) \text{ modulo } 26 \\
 &= (3 - 3) \text{ modulo } 26 \\
 &= 0 \text{ modulo } 26 \\
 &= 0 = A
 \end{aligned}$$

Hasil Dekripsi adalah “DARMA”

8. PHP (*Hypertext Preprocessor*)

PHP merupakan singkatan dari “*PHP: Hypertext Preprocessor*” adalah

sebuah bahasa *scripting* yang terpasang pada *HTML*. Sebaian sintaks mirip dengan bahasa *C*, *Java*, *Asc* dan *Perl*, ditambah beberapa fungsi *PHP* yang spesifik. Tujuan utam adari bahasa ini untuk memungkinkan perancang *web* untuk menulis halaman *web* dinamik dengan cepat.

Berikut adalah tipe data yang ada pada bahasa pemrograman *PHP* adalah sebagai berikut :

a. *String*

Merupakan tipe data berupa sejumlah karakter atau huruf seperti “selamat pagi Darma”. Nilai *string* disisipkan antara tanda kutip satu (‘ ‘) maupun kutip dua (“ “). Contohnya :

```
<? Php
```

```
$x = “selamat pagi Darma”;
```

```
$y = ‘selamat pagi Darma’;
```

```
echo $x;
```

```
echo “<br>”;
```

```
echo $y;
```

```
?>
```

b. *Integer*

Tipe data *integer* adalah angka non-desimal antara -2,147.483.648 dan 2.147.483.647. aturan *integer* untuk tipe data *integer* :

- *Integer* harus memiliki setidaknya satu digit.
- Bilangan bulat tidak boleh memiliki titik desimal.
- *Integer* dapat berupa positif atau negatif.

- Bilangan bulat dapat ditentukan dalam : *desimal* (basis 10), *heksadesimal* (basis 16), *oktal* (basis 8), atau *notasi biner* (basis 2).
Dalam contoh berikut \$ x adalah bilangan bulat. Fungsi PHP *var_dump* () mengembalikan tipe data dan nilai :

```
<?php
$x = 5985;
Var_dump ($x);
?>
```

c. *Float*

Berlawanan dengan *integer*. *Float* adalah tipe data yang mengandung nilai desimal atau berbentuk *eksponensial*. Penting kita ketahui bahwa penulisan angka desimal dalam PHP menggunakan titik (.) bukan koma (,). Dalam contoh berikut variabel \$x adalah *float* :

```
<?php
$x = 10,365;
Var_dump ($x);
?>
```

d. *Boolean*

Tipe data *boolean* mempresentasikan 2 statement yakni : *TRUE* atau *FALSE*, *TRUE* berarti nilai atau valuenya benar, sedangkan *false* berarti salah . umumnya *boolean* digunakan untuk pengujian kondisi.

```
$x = true ;
$y = false;
```

e. *Array*

Array merupakan kode dalam PHP dengan fungsi untuk menyimpan beberapa nilai dalam satu variabel. Jadi kita tidak perlu repot-repot memberikan variabel pada setiap nilai yang akan kita input. Berikut ini contohnya:

```
<?php
$buah = array ("jeruk", "apel", "semangka");
echo "$buah";
?>
```

9. HTML (*Hyper Text Markup Language*)

HTML, atau *Hyper Text Markup Language*, adalah bahasa markup yang digunakan untuk membuat dan mendefinisikan struktur konten pada halaman web. *HTML* adalah komponen dasar dari *World Wide Web* dan merupakan bahasa yang digunakan untuk merancang dan mengatur tampilan elemen-elemen seperti teks, gambar, tautan, formulir, tabel, audio, dan video pada halaman web.

10. XAMPP

Menurut Hilmi Fuad, dkk (2018:2) dalam *jurnal sisfotek global*, "*XAMPP* adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* di komputer lokal. *XAMPP* berperan sebagai *server web* pada komputer. *XAMPP* juga dapat disebut sebuah panel server virtual, yang dapat membantu anda melakukan preview sehingga

dapat memodifikasi *website* tanpa harus online atau terakses dengan internet”.

Berikut ini fitur-fitur XAMPP antara lain :

a. *Apache*

Apache, atau lebih dikenal dengan "*Apache HTTP Server*," adalah perangkat lunak *server web open source* yang sangat populer dan andal. *Apache* awalnya dikembangkan oleh *Tim Berners-Lee* di *CERN* (Organisasi Eropa untuk Penelitian Nuklir) pada tahun 1995 dan sejak itu telah menjadi salah satu *server web* yang paling banyak digunakan di seluruh dunia.

b. *MySQL*

MySQL adalah sistem manajemen basis data (DBMS) *open source* yang sangat populer dan andal. *MySQL* dikembangkan oleh *Oracle Corporation* (sebelumnya oleh *MySQL AB*, kemudian *Sun Microsystems*) dan dirilis sebagai perangkat lunak sumber terbuka, yang berarti sumber kode *MySQL* dapat diakses, dimodifikasi, dan digunakan oleh siapa saja sesuai dengan persyaratan lisensi *open source* yang sesuai.

c. *PHP*

PHP, yang merupakan singkatan dari "*Hypertext Preprocessor*" (sebelumnya, *PHP* adalah singkatan dari "*Personal Home Page*"), adalah bahasa pemrograman skrip yang sering digunakan untuk pengembangan situs web dinamis dan aplikasi web. *PHP* adalah salah satu bahasa pemrograman *server-side* yang paling populer dan banyak digunakan di seluruh dunia.

Bagian-bagian penting XAMPP adalah antara lain :

a. *Htdocs*

Htdocs adalah direktori yang umumnya digunakan dalam *server web Apache* untuk menyimpan file dan dokumen yang akan diakses melalui situs web. Ini adalah singkatan dari "*HyperText Documents*," dan sering kali merupakan direktori utama di mana Anda akan menempatkan file *HTML*, *CSS*, *JavaScript*, gambar, dan konten web lainnya yang akan disajikan melalui *server web*.

b. Control Panel

Control Panel yang biasanya terkait dengan sistem operasi *Windows* adalah untuk mengatur konfigurasi dan pengaturan sistem operasi, bukan terkait langsung dengan pengaturan database, pengunggahan file, atau pengaturan aplikasi seperti *XAMPP*.

c. *PhpMyAdmin*

PhpMyAdmin adalah perangkat lunak administrasi basis data berbasis web yang dirancang khusus untuk mengelola basis data *MySQL*. Ini menyediakan antarmuka web yang memungkinkan pengguna untuk melakukan berbagai tugas administratif terkait basis data *MySQL* melalui *browser web*, tanpa perlu menggunakan baris perintah atau klien basis data terpisah

•

11. Web Server

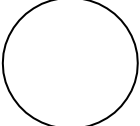
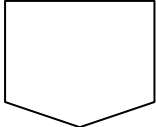
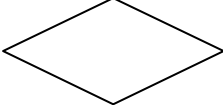
Web server adalah perangkat keras atau perangkat lunak yang bertugas untuk menerima permintaan (*request*) dari klien melalui jaringan, mengelola

permintaan tersebut, dan mengirimkan respon (*response*) kembali kepada klien. *Web server* berperan dalam mengirimkan halaman web, data, dan berkas-berkas lain kepada browser pengguna, yang kemudian menampilkan konten tersebut..

12. Bagan Alir Program

Bagan alir program (*flowchart*) adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan mempresentasikan simbol-simbol tertentu yang mudah dipahami, mudah digunakan dan standar. Tujuan penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan simbol-simbol standar yang dapat dimengerti oleh programmer.

Tabel 2. 2 Simbol flowchart

Simbol	Keterangan
	Proses iiinput iiatau iioutput iiterlepas iidari iijenis iiperangkat
	Proses iioperasional iikomputer
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain
	Awal iiatau iiakhir iisuatu iiprogram (prosedur).
	Koneksi iipenghubung iiproses iikeproses iilain iipada iihalaman iiyang iisama
	Menyatakan iijalannya iiarus iisuatu iiproses
	Simbol keputusan, digunakan unuk suatu penyelesaian kondisi didalam program
	Proses iiinput iiatau iioutput iiterlepas iidari iijenis iiperangkat

13. UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah bahasa visual yang digunakan dalam rekayasa perangkat lunak untuk mendokumentasikan, merancang, dan memodelkan sistem perangkat lunak. *UML* adalah standar industri yang diterima secara luas yang memberikan cara yang terstruktur dan konsisten untuk menggambarkan berbagai aspek sistem perangkat lunak, seperti struktur, perilaku, dan interaksi antar komponen.

UML digunakan secara luas dalam industri perangkat lunak untuk merancang, mengembangkan, dan memahami sistem perangkat lunak yang kompleks. Penggunaan *UML* membantu dalam menghindari kesalahan desain, mempromosikan komunikasi yang lebih baik dalam tim pengembangan, dan menghasilkan dokumentasi yang kuat tentang sistem perangkat lunak. Ini adalah alat yang berharga dalam rekayasa perangkat lunak yang memiliki kompleksitas tinggi. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model.

a. Sesuatu (*Things*)

Ada empat *things* dalam *UML* :

a) *Structural things*

Bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.

b) *Behavioral things*

dalam konteks *Unified Modeling Language* (UML), "*behavioral things*" atau "bagian dinamis" adalah elemen-elemen model yang

digunakan untuk menggambarkan perilaku sistem atau komponen sistem. Salah satu elemen UML yang paling umum digunakan untuk menggambarkan perilaku ini adalah diagram aktivitas.

c) *Grouping things*

pengorganisasian dan pengelompokan elemen-elemen dalam model *UML* sangat penting untuk mengelola kompleksitas dan membuat model lebih mudah dimengerti. Dalam konteks *UML*, penggunaan paket adalah salah satu cara yang efektif untuk mengorganisasi elemen-elemen model yang kompleks, serta mengelompokkan komponen-komponen terkait ke dalam unit yang lebih besar..

d) *An notational things*

elemen-elemen yang digunakan dalam model *UML* untuk memberikan penjelasan, komentar, dan deskripsi lebih lanjut tentang elemen-elemen model. Dalam *UML*, elemen-elemen ini umumnya dikenal sebagai "*annotasi*" atau "*notasi*," dan mereka digunakan untuk meningkatkan pemahaman model dan menjelaskan elemen-elemen model dengan lebih detail.

b. Relasi (*Relationship*)

Relationship (hubungan) dalam *unified modelling language (UML)* antara lain :

a) Keterangan (*Dependency*)

Keterangan (*Dependency*) adalah salah satu jenis hubungan dalam *Unified Modeling Language (UML)* yang menggambarkan hubungan

antara dua elemen dalam model, di mana elemen yang satu (elemen yang bergantung) bergantung pada elemen yang lain (elemen independen) dalam hal perubahan atau penggunaan.

b) Asosiasi

Asosiasi adalah salah satu jenis hubungan dalam *Unified Modeling Language* (UML) yang menggambarkan bagaimana objek atau elemen-elemen dalam model berhubungan satu sama lain.

c) Diagram

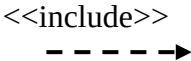
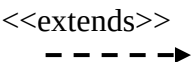
terdapat sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifat statis dan dinamisnya. Dalam pemodelan UML, diagram-dalam diagram ini digunakan untuk menggambarkan berbagai aspek sistem perangkat lunak.

Adapun diagram yang sering digunakan pada penelitian dan perancangan sistem adalah *Use Case Diagram*, *Activiti Diagram*, *Sequence Diagram*.

a. Use Case Diagram

Diagram Kasus Pengguna (Use Case Diagram) adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara pengguna atau aktor dengan sistem perangkat lunak. Diagram ini memberikan gambaran tingkat tinggi tentang fungsionalitas sistem dan skenario penggunaan yang berbeda.

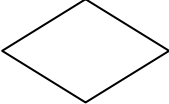
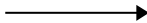
Tabel 2.3 Simbol *Usecase diagram*

Simbol	Nama	Keterangan
	Aktor	Aktor adalah entitas eksternal yang berinteraksi dengan sistem.
	<i>Use Case</i>	Kasus pengguna adalah suatu fungsi, fitur, atau tindakan yang dilakukan oleh sistem yang memberikan nilai kepada satu atau lebih actor
	<i>Relasi Asosiasi</i>	Garis solid yang menghubungkan aktor dengan kasus pengguna mengindikasikan bahwa aktor tersebut terlibat dalam skenario penggunaan kasus pengguna tersebut
	<i>Relasi Include</i>	Memungkinkan suatu use case menggunakan fitur yang dimiliki oleh use case lain Memungkinkan suatu use case secara pilihan menggunakan fitur yang dimiliki oleh use case lain
	<i>Relasi Extend</i>	Extend digunakan untuk menggambarkan perluasan di mana satu kasus pengguna dapat "memperpanjang" atau meluaskan fungsionalitas kasus pengguna lain

b. *Activity Diagram*.

Diagram ini bersifat dinamis. Diagram ini juga merupakan tipe yang khusus dari suatu diagram status dengan memberikan gambaran aliran yang berasal dari suatu aktivitas ke aktivitas yang lain pada sistem. Pentingnya diagram aktivitas ini terhadap pemodelan fungsi-fungsi yang terdapat pada suatu sistem.

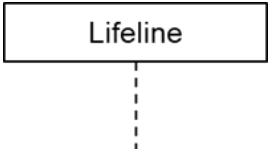
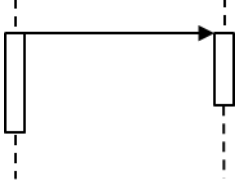
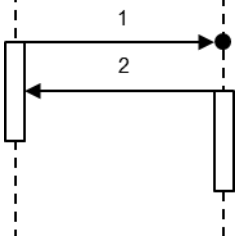
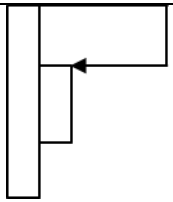
Tabel 2.4 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start state</i>	Titik strat
	<i>End state</i>	Titik akhir
	<i>Activity</i>	Aktivitas yang dilakukan aktor
	<i>Decision</i>	Memungkinkan suatu use case untuk menggunakan fungsionalitas yang dilakukan oleh use case lainnya.
	<i>Interaction</i>	Alur

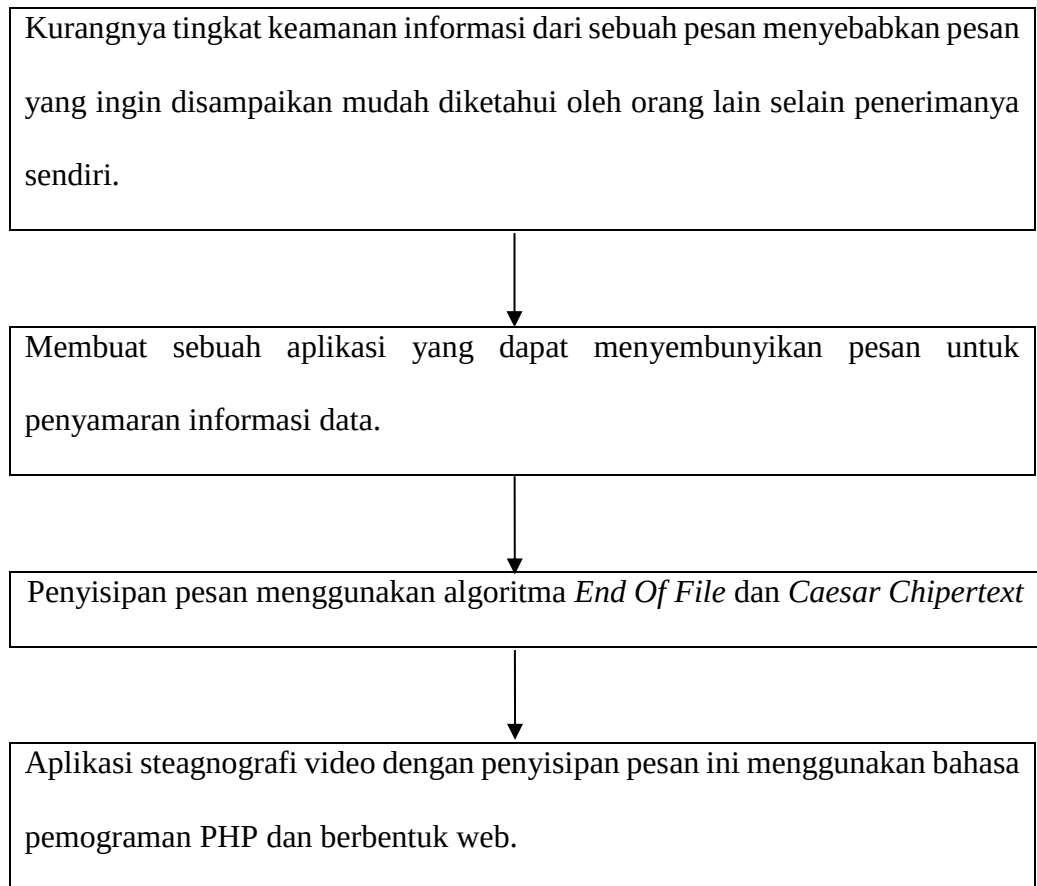
c. Sequence Diagram

Diagram Urutan (*Sequence Diagram*) adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara objek-objek dalam sistem sepanjang waktu. Diagram urutan menggambarkan urutan pesan atau panggilan metode yang dikirim antara objek-objek dalam sebuah skenario atau *use case* tertentu. Diagram ini membantu dalam memodelkan bagaimana objek berinteraksi dan berkomunikasi selama eksekusi.

Tabel 2.5 *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Lifeline</i>	Objek dalam diagram urutan mewakili entitas atau instance yang terlibat dalam interaksi.
	<i>Create Message</i>	Pesan-pesan ini menggambarkan urutan komunikasi antara objek-objek.
	<i>Synconous Message</i>	Mengirimkan pesan ini mengaktifkan sebuah proses; hanya setelah proses tersebut selesai, Anda dapat mengirimkan pesan baru..
	<i>Message to self</i>	Suatu hasil kembalian sebuah operasi dan berjalan pada objek itu sendiri.

14. Kerangka Pikir



Gambar 2.3. Kerangka Pikir

B. Kajian Hasil Penelitian Terdahulu

1. Penelitian yang dilakukan oleh Nurrahmi Wahyuni pada tahun 2022 dengan judul *Aplikasi Steganography Pdf to Image Dengan Metode Spread Spectrum*. Pada penelitian ini penulis membuat aplikasi enkripsi dan dekripsi yang dapat digunakan untuk menyembunyikan informasi. Metode algoritma yang digunakan adalah *spread spectrum*. Berdasarkan pengujian yang dilakukan dihasilkan sebuah aplikasi yang dapat menyembunyikan informasi dengan menggunakan gambar sebagai penyembunyi. Hasil dari penelitian ini adalah berupa perubahan ukuran gambar sesudah disisipkan file *pdf*, dan perubahannya tergantung dari hasil *spread spectrum*-nya.
2. Penelitian yang dilakukan oleh Andi Herul pada tahun 2019 dengan judul *Aplikasi Steganografi Menggunakan Metode LSB (Least Significant Bit) Pada Media Audio*. Pada penelitian ini penulis membuat aplikasi steganografi yaitu dengan menyembunyikan file teks pada *cover audio* *war*. Penelitian ini menghasilkan aplikasi yang dapat menyembunyikan informasi file teks dengan *cover audio* dengan ukuran file audio, *bit rate*, dan panjang audionya sama sesuai dengan aslinya. Metode yang digunakan *LSB*.
3. Penelitian yang dilakukan oleh Selpi pada tahun 2021 dengan judul *Aplikasi Perpaduan Enkripsi Algoritma Base64 Dengan Metode Steganografi Discrete Cosine Transform (DCT)*. Pada penelitian ini penulis membuat aplikasi *enkripsi* teks dengan algoritma *base64* lalu disisipkan ke gambar dengan metode *steganografi* *DCT*.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini menggunakan penelitian dimana *eksperimental* adalah jenis penelitian di mana peneliti melakukan percobaan atau eksperimen untuk mengumpulkan data secara langsung dari hasil percobaan yang dilakukan. Penelitian *eksperimental* biasanya melibatkan manipulasi *variabel* bebas (*independen*) untuk mengukur dampaknya terhadap variabel terikat (*dependen*) dalam rangka mengidentifikasi hubungan sebab-akibat..

B. Jadwal Penelitian

Tabel 3.1 Jadwal pelaksanaan penelitian

No	UraianKegiatan	Bulan 2023	
		AGUSTUS	SEPTEMBER
1	Studi literature		
2	Analisis data		
3	Perancangan aplikasi		
4	Pembuatan aplikasi		
5	Pengujian aplikasi		

C. Alat dan Bahan Penelitian

1. Perangkat Keras

Perangkat keras yang digunakan adalah sebuah laptop dengan spesifikasi yang dapat dilihat dari tabel berikut :

Tabel 3.2 Perangkat Keras

Jenis	Spesifikasi
Laptop	ACER
Processor	Intel(R) Celerom(R) CPU N3050 160Hz
Memory	2 GB
Hardisk	500GB

2. Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun aplikasi ini dapat dilihat pada tabel berikut :

Tabel 3.3 Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemrograman	PHP
Tools	XAMPP
	Sublime Text
	Browser Google Chrome

3. File Video dan Pesan Teks

Bahan yang digunakan dalam penyisipan pesan ini adalah file video MP4, dengan kapasitas 600 MB

D. Teknik Pengumpulan Data

Pada metode pengumpulan data untuk mendapatkan data yang cukup akurat, maka digunakan beberapa metode dibawah ini yaitu:

1. Studi Pustaka

Didalam memperoleh data-data yang dibutuhkan, maka penulis membaca dan mempelajari buku-buku, artikel, jurnal, dan skripsi yang terkait dengan metode *End Of File (EOF)* dan metode *Caesar Chipertext*.

2. Eksperimen

Eksperimen adalah tahap penelitian yang penting dalam metode ilmiah. Dalam eksperimen, peneliti merancang dan melakukan serangkaian percobaan atau pengujian untuk mengumpulkan data secara langsung. Tujuan eksperimen adalah untuk menguji hipotesis dan mendapatkan pemahaman yang lebih mendalam tentang hubungan sebab-akibat atau fenomena yang sedang diteliti.

E. Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah sebagai berikut :

1. Pengumpulan Data

Dalam konteks penelitian ini tentang steganografi video menggunakan metode *End Of File* dan *Caesar Chipertext*, pengumpulan data mungkin melibatkan informasi tentang teknik *steganografi*, karakteristik video, dan algoritma yang akan digunakan.

2. Analisis Data

Setelah data terkumpul, peneliti melakukan analisis data untuk mengidentifikasi pola, tren, atau informasi yang relevan.

3. Perancangan Aplikasi

Pada tahap ini mencakup pembuatan *flowchart*, penggunaan *UML* untuk merancang struktur aplikasi, dan desain antarmuka.

4. Pembuatan Aplikasi

Pada tahap ini, peneliti mulai membangun aplikasi berdasarkan perancangan yang telah dibuat. Bahasa pemrograman PHP digunakan untuk mengimplementasikan algoritma *End Of File* dan *Caesar Chipertext*.

5. Pengujian Aplikasi

Aplikasi yang telah dibangun kemudian diuji untuk memastikan keberhasilannya dalam melakukan penyisipan teks ke dalam video.

6. Implementasi

Setelah aplikasi diuji dan dianggap berhasil, langkah selanjutnya adalah mengimplementasikan aplikasi ini secara luas atau sesuai dengan keperluan.

F. Metode Pengujian

Penting untuk melakukan sejumlah kasus uji dengan berbagai taktik, pertanyaan, atau rute navigasi yang mencerminkan penggunaan sistem yang umum, penting, atau tidak biasa. Memilih serangkaian kasus uji yang tepat secepat dan sekecil mungkin untuk memastikan perilaku sistem secara detail merupakan langkah penting dalam pengembangan sistem. Pengujian unit, yang memverifikasi validitas proses dan beroperasi secara independen dari komponen sistem lainnya, harus menjadi bagian dari pengujian. Selanjutnya, pengujian modul perlu dilakukan untuk menentukan apakah eksekusi beberapa modul terkait, serta apakah kombinasi beberapa unit dalam satu modul, beroperasi sebagaimana mestinya dan memenuhi persyaratan sistem.

Pengujian unit sama pentingnya jika struktur kontrol antar modul telah terbukti baik. pengujian unit Pengujian unit digunakan untuk menguji setiap modul untuk memastikan setiap modul menjalankan fungsinya sebagaimana mestinya. Ada dua metode untuk melakukan unit testing yaitu :

1. *White Box Testing*

Lihatlah pengujian *white box*, teknik desain kasus uji yang menghasilkan kasus uji menggunakan struktur. Tes ini digunakan untuk memperkirakan operasi perangkat lunak secara rinci berdasarkan jalur logika. Pengulangan dan kondisi fisik digunakan untuk menguji perangkat lunak. Seluruh proses pada aplikasi akan diuji dengan metode *black box* untuk melihat apakah perangkat lunak dapat berfungsi, input diterima dengan baik, dan output dihasilkan dengan cepat. Pengujian *black box* merupakan pengujian yang bertujuan untuk menunjukkan fungsi perangkat lunak dan cara kerja program. Ketika pengguna memasukkan kata sandi pengguna yang salah, misalnya, pengujian kotak putih akan memperingatkan mereka dan memberikan informasi kepada

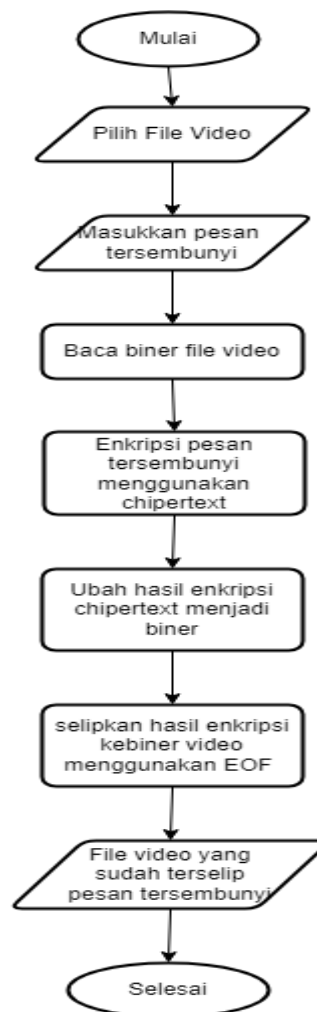
2. *Black Box*

Prosedur dan tombol navigasi dan instalasi program lain tampaknya bebas dari kesalahan, menurut temuan tes. Sangat penting untuk mematuhi aturan yang ditetapkan oleh program, karena tidak melakukannya akan mengakibatkan sistem menolak perintah yang tidak tepat. Ketika pengguna memasukkan data dengan salah, misalnya, sistem dapat memberikan informasi kepada pengguna sebagai hasil dari pemeriksaan mekanisme perangkat lunak, yang memeriksa jalur dasar atau proses looping program dan mengevaluasi efektivitas desain aplikasi.

G. Desain Sistem

1. Flowchart Sistem

a. Proses *Encoding* : Penyisipan Pesan

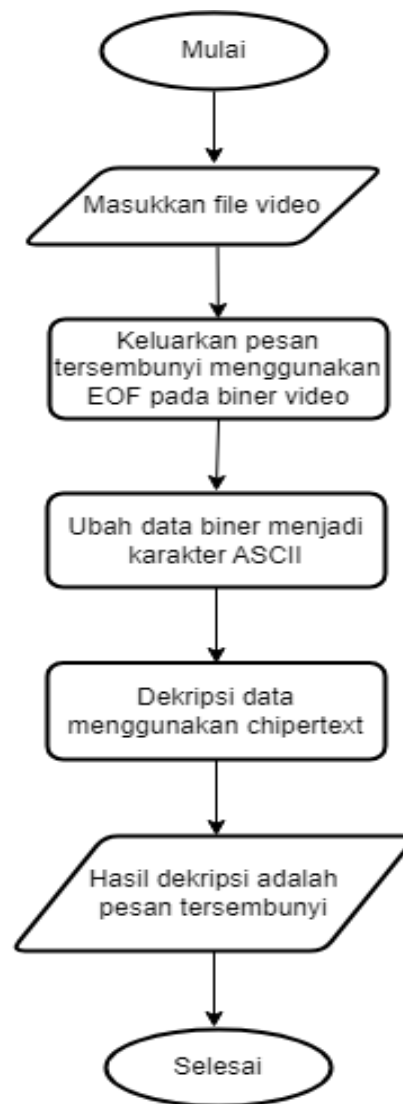


Gambar 3.1 Flowchart proses encoding pesan

Flowchart pada gambar 3.1 menjelaskan bagaimana pengguna memilih file video, membaca biner yang terdapat dalam file video, dan kemudian menggunakan metode *ciphertext* untuk mengenkripsi pesan. Pengguna kemudian mengubah pesan terenkripsi tersebut ke dalam bentuk biner, menyisipkan hasil enkripsi biner video tersebut

menggunakan EOF, dan berakhir dengan pesan tersembunyi di dalam file video tersebut.

b. Proses *Decoding* : pengambila pesan



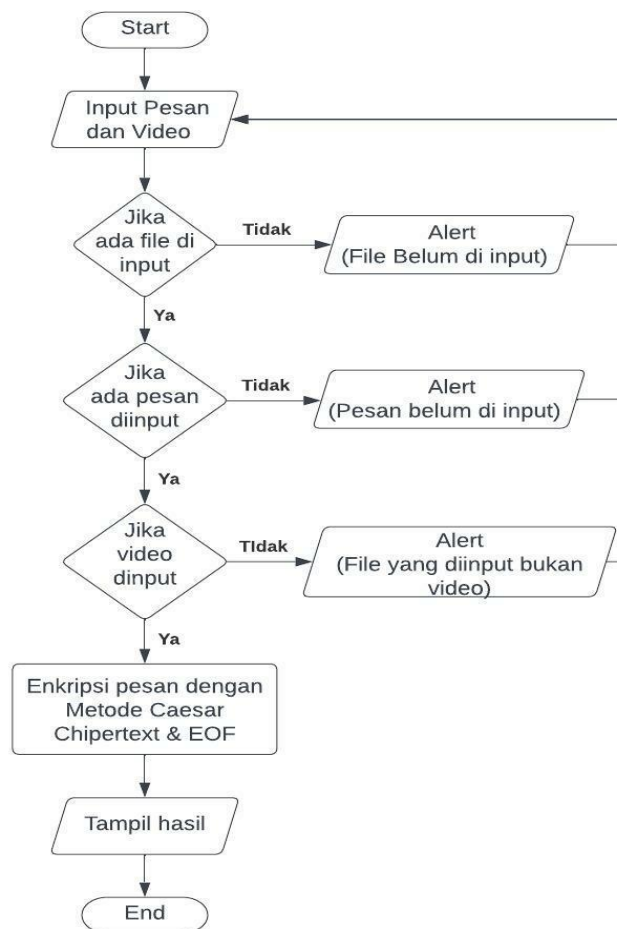
Gambar 3.2 Flowchart proses decoding pesan

Pada gambar 3.2 menjelaskan bagaimana pengguna memasukkan file video, menggunakan EOF untuk menghapus pesan tersembunyi dari biner video, dan kemudian melanjutkan untuk memecahkan kode

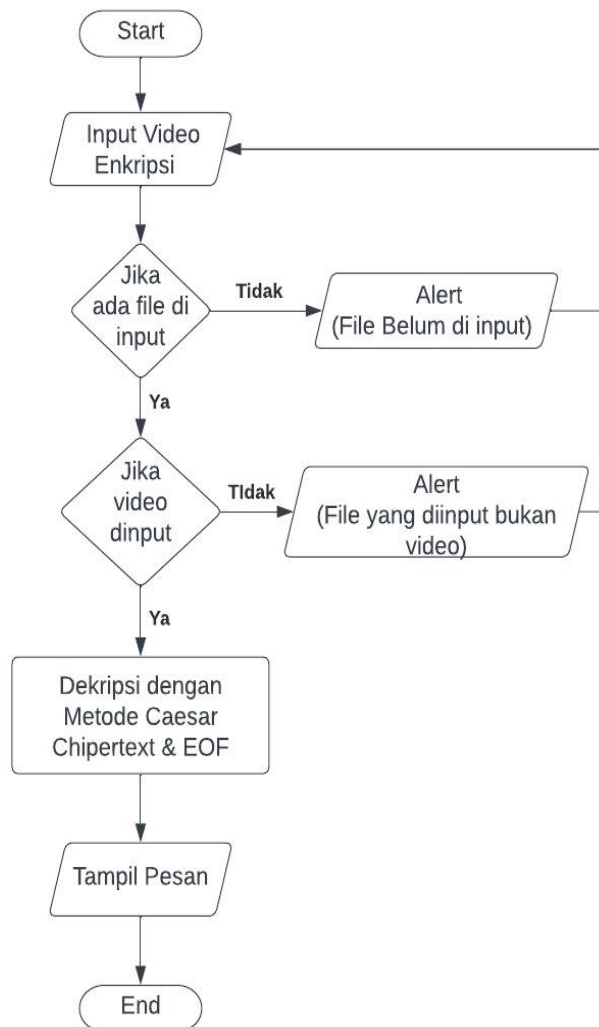
pesan atau mengambil pesan asli. Data biner kemudian harus dikonversi ke karakter ASCII. Selanjutnya, gunakan *ciphertext* untuk mendekripsi data; hasil dekripsinya adalah pesan tersembunyi.

2. Flowchart Aplikasi

a. Proses Enkripsi



Gambar 3.3 Proses Enkripsi

b. Proses Dekripsi**Gambar 3.4** Proses Dekripsi

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Perancangan Sistem

Aplikasi yang dikembangkan pada tugas akhir ini pada dasarnya adalah aplikasi *steganografi* video yang menyisipkan teks menggunakan metode *Chipertext* dan *End Of File*. Ini terdiri dari perangkat keras dan perangkat lunak sebagai dua komponen utamanya, yang berarti bahwa desain perangkat keras dan perangkat lunak membentuk desain sistem.

1. Rancangan Perangkat Keras

Adapun perangkat keras yang digunakan untuk mengimplementasikan rancangan pada penelitian ini adalah berupa laptop *ACER* dengan spesifikasi *Intel (R) Celerom (R) CPU N3050 160Hz, RAM 2 GB*.

2. Rancangan Perangkat Lunak

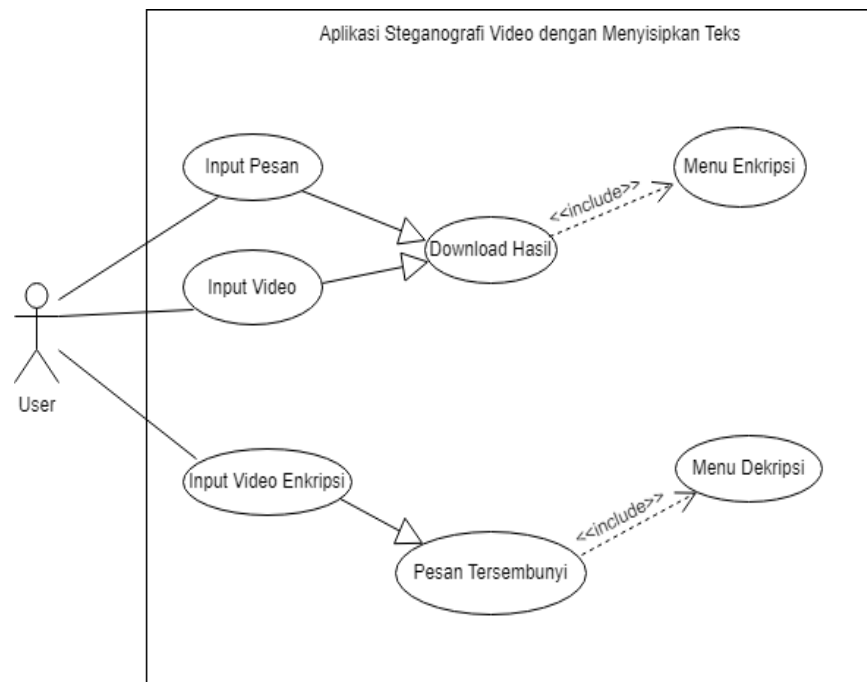
Adapun perangkat lunak yang digunakan untuk mengimplementasikan rancangan pada penelitian ini adalah rancangan bangun perangkat lunak yaitu menggunakan sistem *operasi windows 10 Pro 64 bit*, bahasa pemrograman *PHP*, dan *tools* seperti *xampp*, *SublimeText*, *Browser* serta jenis file yang digunakan saat penelitian adalah teks dan video. Kemudian untuk perancangan sistem, perlu dilakukan agar memberikan gambaran yang jelas dan lengkap tentang rancangan bangun dan implementasi bagaimana sistem dibuat terdiri dari :

a. Perancangan Model UML (*Unified Modelling Language*)

Sistem yang penulis rancang adalah sistem berupa penyisipan pesan kedalam media cover yaitu video. Adapun dalam melakukan perancangan sistem ialah dengan memanfaatkan diagram *UML* berupa *Usecase diagram*, *Activity diagram*, *Sequence diagram*.

1. Usecase Diagram

Usecase diagram berfungsi untuk menjelaskan alur sistem jika dilihat menurut pandangan orang yang berada diluar sistem.



Gambar 4. 1 Usecase diagram

Penjelakasan *usecase diagram* :

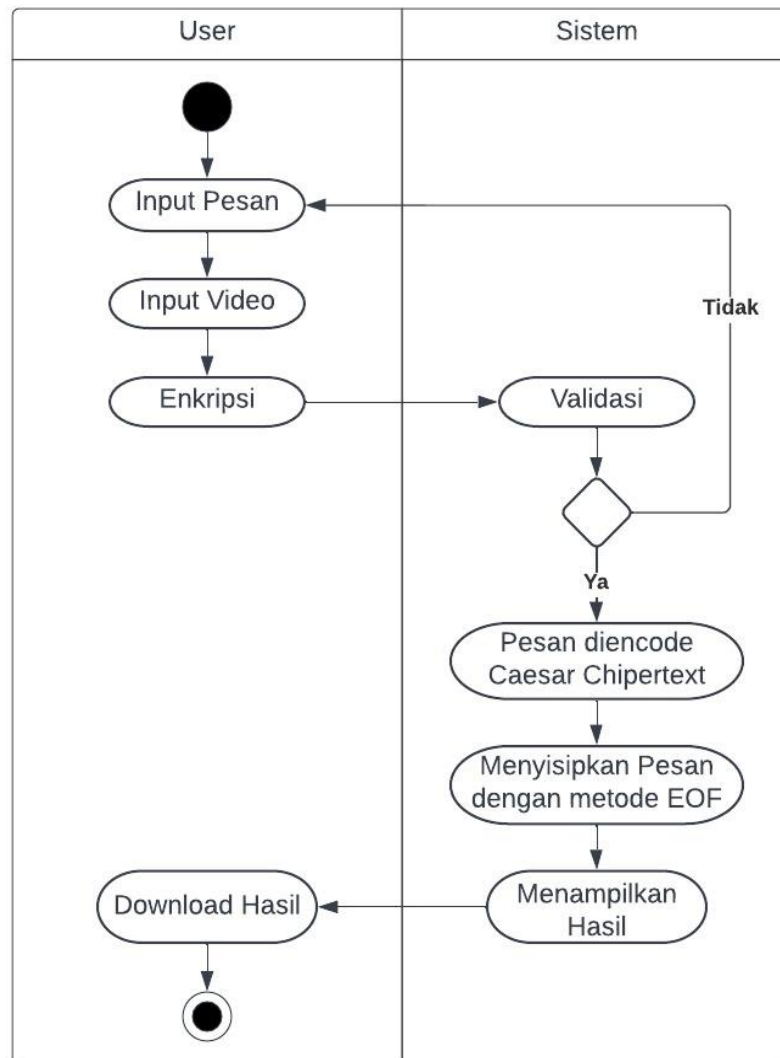
Tabel 4.1 Penjelasan *usecase* diatas

Nama use case	Deskripsi <i>use case</i>
Input pesan	<i>Use case</i> ini menjelaskan user bisa menginput pesan terlebih dahulu .
Input video	<i>Use case</i> ini menjelaskan user menginput video.
Download hasil	<i>Use case</i> ini menjelaskan user memilih download video yang telah disisipkan teks.
Menu <i>Enkripsi</i>	<i>Use case</i> ini menjelaskan tentang proses <i>Enkripsi</i> menggunakan metode <i>Caesar Chipertext</i> dan metode <i>End Of File</i> .
Input video <i>enkripsi</i>	<i>Use case</i> ini menjelaskan user bisa menginput video yang sudah didownload yang ingin di <i>Dekripsi</i> .
Menu <i>Dekripsi</i>	<i>use case</i> ini menjelaskan proses <i>Dekripsi</i> video dengan menggunakan metode <i>Caesar Chipertext</i> dan metode <i>End Of File</i> .

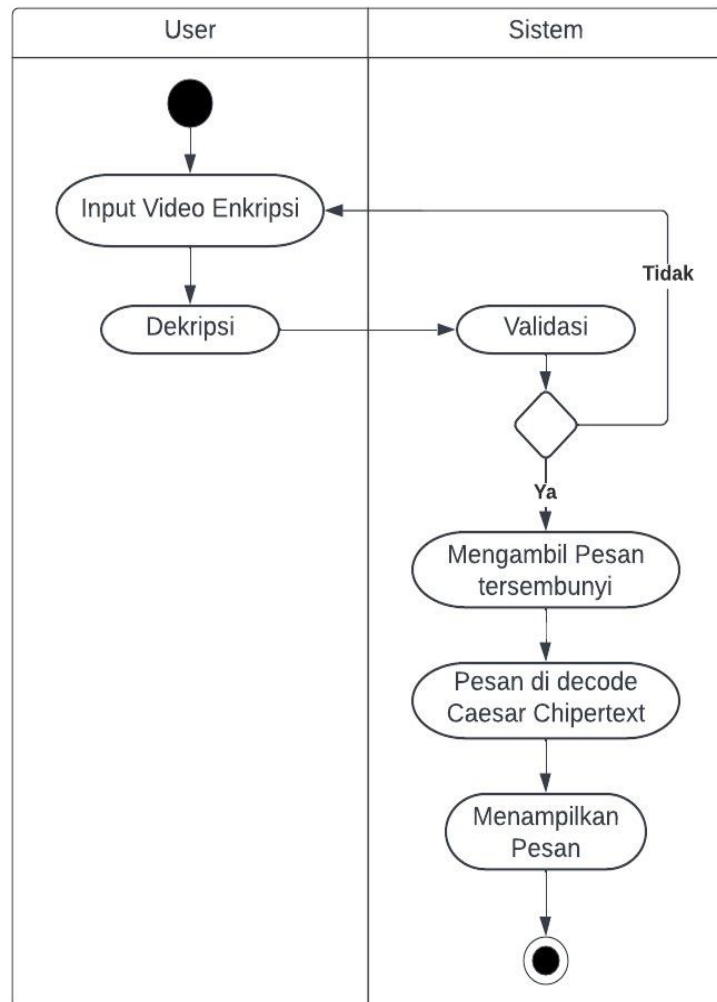
2. Activity Diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses dan urutan dalam suatu proses. *Activity diagram* pada sistem yang penulis rancang antara lain :

a) Activity Diagram Enkripsi

**Gambar 4.2** Activity diagram enkripsi

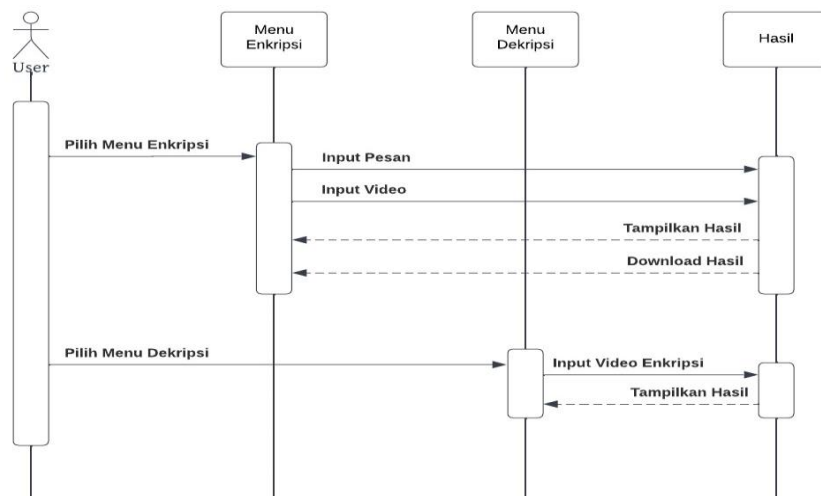
b) Activity Diagram Dekripsi



Gambar 4.3 Activity diagram dekripsi

3. Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antara objek dan mengkondisikan komunikasi antara objek-objek tersebut. Berikut *sequence diagram* aplikasi yang akan dibangun.



Gambar 4.4 Sequence diagram

B. Perancangan Aplikasi

Pada perancangan aplikasi, aplikasi akan memanfaatkan fitur *shortcut* pada *google chrome* agar *browser* berjalan konstan atau berjalan dengan satu tab. Adapun rancangan tampilan yaitu :

1. Tampilan Halaman Utama

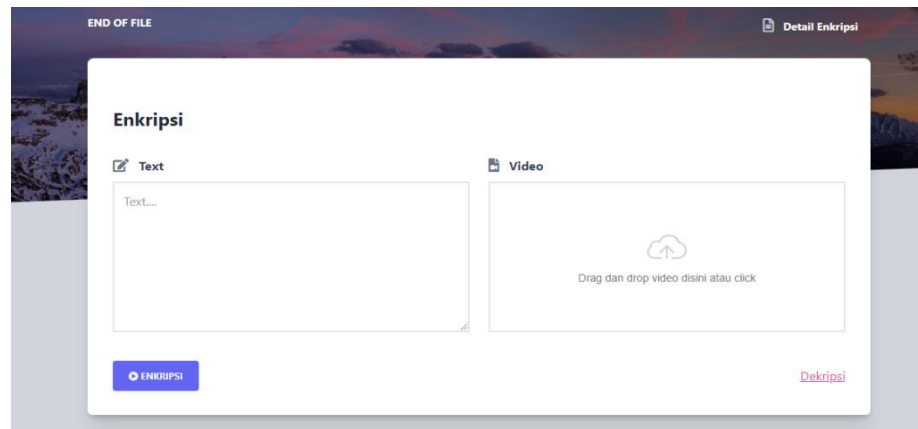
Pada tampilan ini merupakan tampilan awal pada saat kita membuka aplikasi, disini terdapat nama aplikasi *vMessage*, judul aplikasi, mulai



Gambar 4.5 Tampilan Halaman Utama

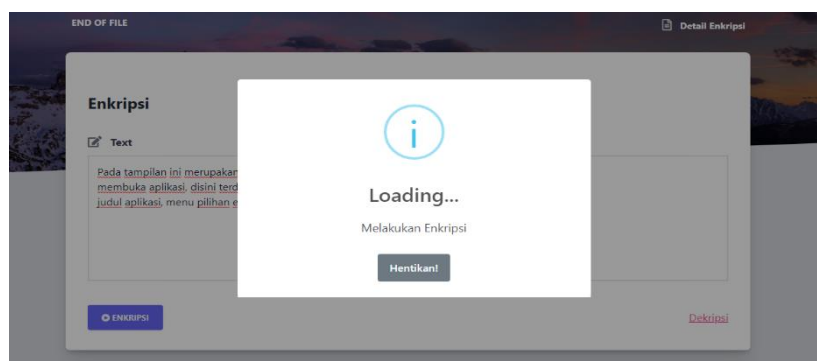
2. Tampilan Menu Enkripsi

Pada halaman ini menampilkan 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.

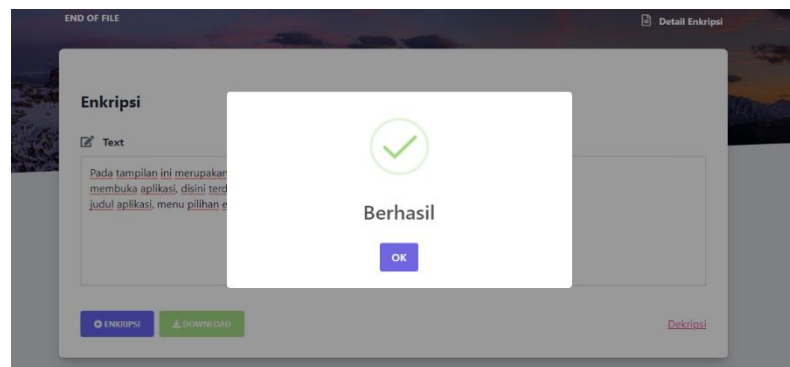


Gambar 4.6 Tampilan Menu Enkripsi

Pada tampilan gambar di bawah menjelaskan bahwa setelah teks dan file video di masukkan maka langkah selanjutnya yaitu user akan melakukan enkripsi maka akan muncul seperti gambar 4.10 dan setelah proses enkripsi itu selesai maka akan muncul pada gambar 4.11



Gambar 4.7 Tampilan Proses Enkripsi



Gambar 4.8 Proses enkripsi

3. Proses Enkripsi

a. Langkah pertama

Pada langkah ini video dan teks yang ingin disembunyikan di-input dan ditampilkan dalam bentuk *hexadesimal* agar lebih ringkas dan mudah dibaca.

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]: Run

Choose file _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 4.9 Proses enkripsi langkah 1

b. Langkah kedua

Pada langkah ini adalah proses *Caesar Chipertext* 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.

Langkah 2

Pada langkah ini adalah proses Caesar Chiptext pada Text yang ingin diselipkan agar text 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



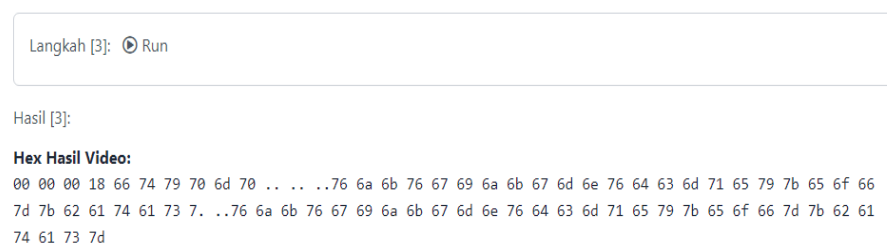
Gambar 4.10 Proses enkripsi langkah kedua

c. Proses enkripsi langkah ketiga

Langkah terakhir pada proses enkripsi ini adalah proses *End Of File* dimana *Caesar Chiptext* akan diselipkan pada akhir video, agar mengetahui batas antara video dan text yang disembunyikan, maka ditentukan “7b 65 6f 66 7d 62 61 74 61 73 7d” sebagai pembatas bentuk text dari bilangan hex tersebut adalah *{eof}{batas}*

Langkah 3

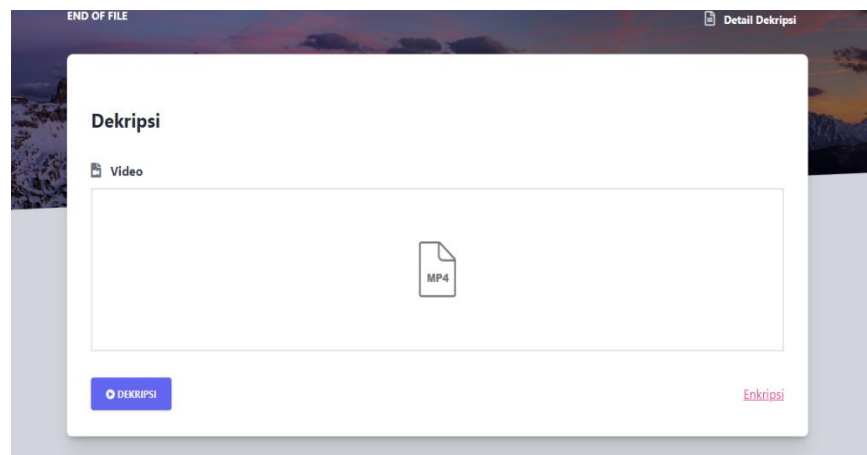
Langkah terakhir pada proses ini adalah proses End of File dimana Caesar Chiper Text akan diselipkan pada akhir Video. Agar mengetahui batas antara video dan text yang disembunyikan, maka ditentukan “7b 65 6f 66 7d 7b 62 61 74 61 73 7d” sebagai pembatas. bentuk text dari bilangan hex tersebut adalah “{eof}{batas}”.



Gambar 4.11 Proses enkripsi langkah ketiga

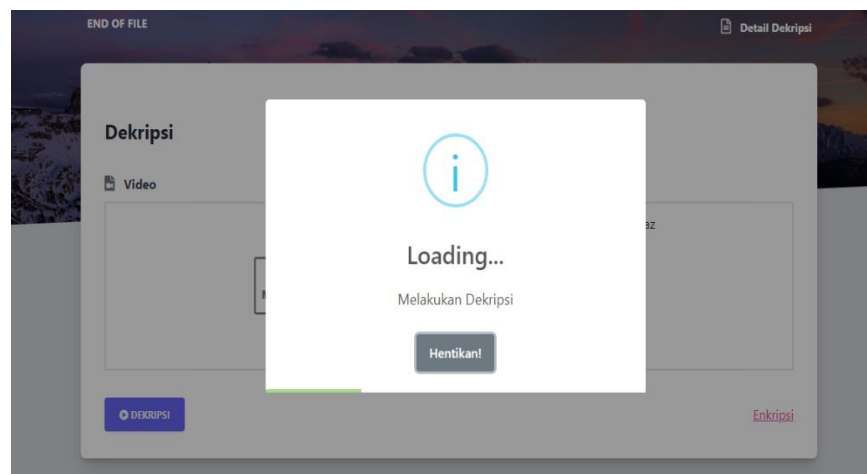
4. Tampilan Menu Dekripsi

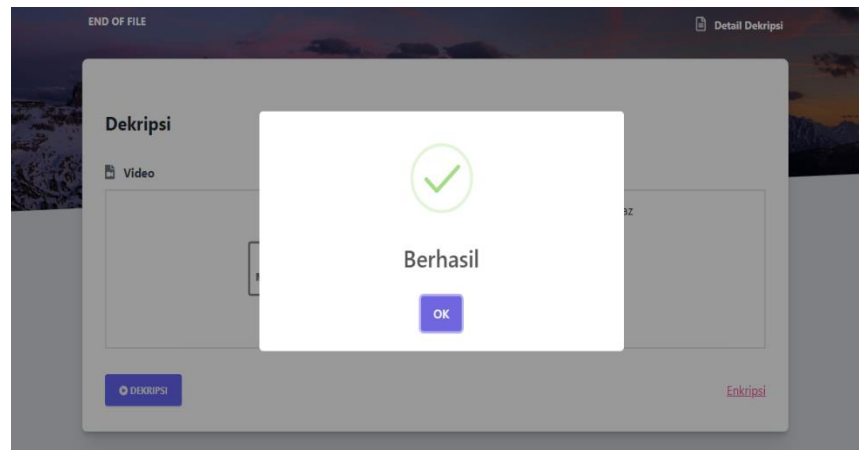
Pada halaman ini user dapat memasukkan video yang sudah berisi pesan atau sudah terenkripsi.



Gambar 4.12 Tampilan Menu Dekripsi

Pada tampilan gambar di bawah menjelaskan bahwa setelah file video di masukkan maka langkah selanjutnya yaitu user akan melakukan dekripsi maka akan muncul seperti gambar 4.10 dan setelah proses dekripsi itu selesai maka akan muncul pada gambar 4.11

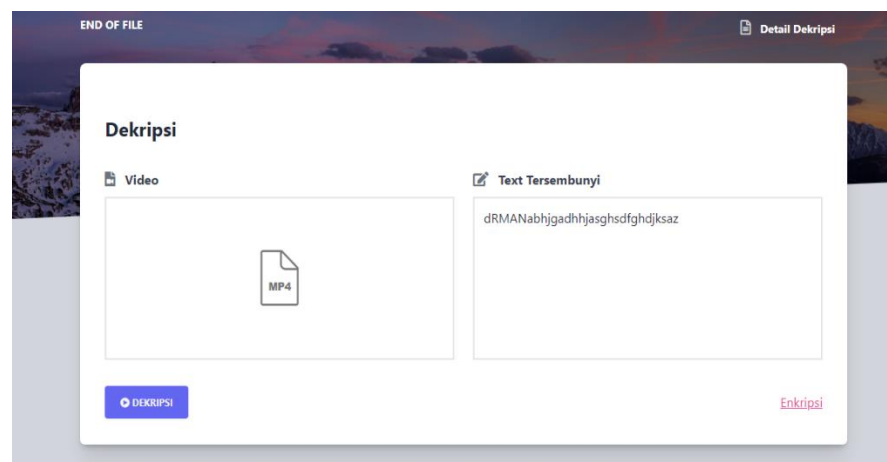




Gambar 4.13 Tampilan Menu Dekripsi Berhasil

5. Hasil Dekripsi

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



Gambar 4.14 Tampilan Hasil Dekripsi

6. Proses Dekripsi

a. Langkah pertama

Pada langkah ini video yang telah diselipkan text tersembunyi di-input dengan ketetapan '7b 65 6f 66 7d 7b 62 61 74 61 73 7d' sebagai pembatas atau dalam bentuk text adalah $\{eof\}$ {batas}, maka

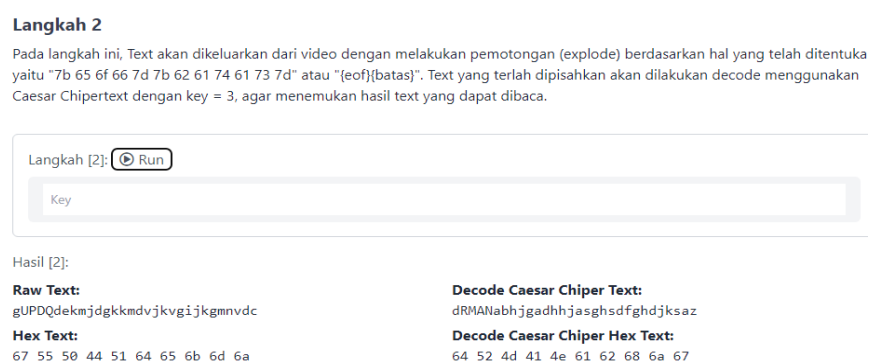
jika batas yang telah ditentukan tidak ada diakhir file maka video tersebut tidak pernah dilakukan proses penyelipan.



Gambar 4.15 Proses dekripsi langkah pertama

b. Langkah kedua

Pada langkah ini, text akan dikeluarkan dari video dengan melakukan pemotongan (*explode*) berdasarkan hal yang telah ditentukan yaitu ‘7b 65 6f 66 7d 7b 62 61 74 61 73 7d’ atau “{eof}{batas}”, text yang telah dipisahkan akan dilakukan *decode* menggunakan *Caesar Chiphertext* dengan *key* = 3 agar menemukan hasil text yang dapat dibaca.



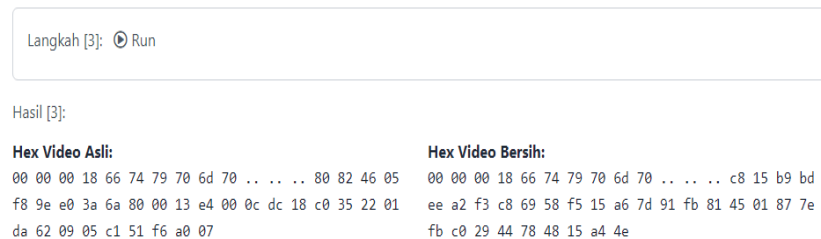
Gambar 4.16 Proses dekripsi langkah kedua

c. Langkah ketiga

Pada langkah ini adalah langka untuk membuat kembali video text tersembunyi telah dikeluarkan agar menjadi video bersih dari text tersembunyi. Keunggulan dari metode *End Of File* adalah memungkinkan menyelipkan text kedalam video tanpa mempengaruhi kualitas *visual video*.

Langkah 3

Pada langkah ini adalah langkah untuk membuat kembali video yang text tersembunyi telah dikeluarkan, agar menjadi video bersih dari text tersembunyi. Keunggulan dari metode End of File adalah mungkin menyelipkan text kedalam video Tanpa mempengaruhi kualitas visual video.



Gambar 4.17 Proses dekripsi langkah ketiga

C. Pengujian Algoritma

Pada tahapan 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

1. Enkripsi

a. 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 diproses.

b. 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.

c. Membuat file hex untuk video

Proses ini untuk mengkonversi biner video kebentuk *hexadesimal*, sebagai representasi dari file video asli yang belum pernah diproses.

d. Membuat file hex untuk text tersembunyi

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

2. Enkripsi Metode *Caesar Chipertext*

a. Membuat file konversi text tersembunyi ke *chipertext*

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

b. 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*.

3. Enkripsi Metode *End Of File*

a. Melakukan penyelipan

Setelah file sudah siap yang dibutuhkan yaitu text tersembunyi yang telah dikonversi ke *chipertext* dan video sebagai wadahnya, selanjutnya adalah menyelipkannya menggunakan *metode EOF*, text yang akan membatasi kedua data tersebut adalah “*{eof}{batas}*” dimana bentuk hexadesimalnya adalah “7B 65 6F 66 7D 7B 62 61 74 61 73 7D”. Setelah digabung akan disimpan kedalam file yang akan menjadi representasi dari hasil penyelipan.

b. Membuat video hasil akhir

Proses ini adalah proses pembuatan video dengan mengkonversi file hexadesimal proses sebelumnya menjadi bentuk biner. Proses akan menghasilkan video yang dapat dijalankan dan telah diselipkan pesan tersembunyi. Adapun proses dekripsi memiliki tiga proses yaitu :

1. Dekripsi

a. Membuat file video

Proses ini memindahkan video yang diinput ke folder aplikasi untuk nantinya diproses, sebagai file awal yang akan di dekripsi.

b. Membuat file *hexadesimal* video

Proses ini adalah proses untuk mengkonversi video yang diinput menjadi hexadecimal, sebagai representasi bentuk *hexadesimalnya*.

2. Dekripsi

a. Mengeluarkan text tersembunyi

Proses ini adalah proses mengeluarkan text tersembunyi ada didalam video. Isi video dalam bentuk *hexadesimal* akan di cek apakah terdapat pembatas didalamnya yaitu “*{eof}{batas}*” atau dalam bentuk *hexadesimalnya* “7B 65 6F 66 7D 7B 62 61 74 61 73 7D” Jika pembatas itu ada didalam video. Jika pembatas itu ada maka text tersembunyi ada didalam video, jika tidak ada maka proses akan dihentikan tetapi jika ditemukan program akan mengambil *hexadesimal* setelah pembatas tersebut dan akan menyimpannya sebagai *hexadesimal* dari text yang terselipkan.

b. Bentuk text dari *hexadesimal* hasil potongan

Setelah mengambil potongan *hexadesimalnya* maka proses ini akan mengkonversi hexadecimal tersebut kedalam bentuk text agar bisa dibaca, sebagai representasi bentuk karakter dari *hexadesimal* yang ditemukan

c. Dekripsi *chipertext*

Setelah ditampilkan karakter text yang diselipkan maka akan ditemukan sebuah *text uniq*. *Text uniq* ini adalah hasil dari enkripsi *chipertext* yang saat enkripsi penyelipan dilakukan. Maka untuk dapat melihat text yang sebenarnya maka akan dilakukan dekripsi *chipertext* dengan *key* yang telah diinput. *Key* ini harus sama dengan *key* yang dimasukkan saat enkripsi *chipertext* . untuk default aplikasinya menggunakan *key* = 3. Setelah

dilakukan dekripsi *chipertext* maka akan mendapatkan hasil text sesuai dengan yang diselipkan

d. Membuat *hexadesimal* dari text yang diselipkan

Proses ini mengkonversi text yang telah didekripsi *chipertext* ke *hexadesimal* dari text yang telah dibuat

3. Dekripsi

a. Mengambil *hexadesimal* video

Sebelumnya telah dibuat file *hexadesimal* video yang diinputkan di dekripsi. File tersebut diproses kembali dan mengambil *hexadesimal* sebelum pembatas. Pada proses ini yang diambil ada sebelum pembatas sebagai *hexadesimal* asli dari video tanpa *hexadesimal* text yang diselipkan.

b. 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 extensi video yang diinputkan pada proses dekripsi ini.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

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 Chiphertext* menggunakan *plaintext* dilihat pada contoh berikut :

a. Tahapan Enkripsi

1. Input

Langkah proses enkripsi *chiphertext* yaitu :

Plaintext : LOVE

K : 3

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

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

Enkripsi : ORYH

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

Bit : 01001111 01010010 01011001 01001000

Deretan binari dijadikan satu baris panjang

“ORYH” : 01001111010100100101100101001000

2. Proses

Bit : 01001111 01010010 01011001 01001000

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 *Cesar Chipertext* menjadi ORYH.

b. Tahapan Dekripsi

Langkah proses dekripsi *chipertext* yaitu :

Plaintext : ORYH

K : 3

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

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

Untuk proses deskripsi atau *decoding* tidak jauh berbeda dengan proses *encoding*,
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 teks akan balik ke semula yaitu LOVE.

4. End Of File (EOF)

a. 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 dengan text tersebut adalah “{eof}{batas}” dimana bentuk *hexadesimalnya* adalah “7B 65 6F 66 7D 7B 62 61 74 61 73 7D”.

20 66 74 79 70 69 73 6F 6D 20 20 02 20 69 73 6F 6D 69 73 6F 32 61 76 63 31
6D 70 34 31 20 20 20 08 66 72 65 65 20 2B BB 2122 6D 64 61 74 20 20 201D
C3 65 B8 40 F7 FF FE 1E 160 B3 20 29 2019 EE 20 29 BC 66 20 29 E6 2B 20
2A 10 07 20 2A 39 DD 20 2A 63 77 20 2A 8D 4F 20 2A B6 DA 20 2A E0 D8
20 2B 0A 69 20 2B 34 6B 20 2B 5D E7 20 2B 2021 B9 20 20 20 62 75 64 74 61
20 20 20 5A 6D 65 74 61 20 20 20 20 20 20 20 20 21 68 64 6C 72 20 20 20 20 20
20 20 20 6D 64 69 72 61 70 70 6C 20 20 20 20 20 20 20 20 20 20 20 2D 69
6C 73 74 20 20 20 25 A9 74 6F 6F 20 20 20 1D 64 61 74 61 20 20 20 01 20 20
20 20 4C 61 76 66 35 37 2E 31 39 2E 31 30 30

Tabel 4.2 Video sebelum disisipkan text

Setelah hasil *chipertext* diubah menjadi *hexadesimal* selanjutnya pesan akan disembunyikan pada akhir file video dengan nilai 4F 52 59 48

20 66 74 79 70 69 73 6F 6D 20 20 02 20 69 73 6F 6D 69 73 6F 32 61 76 63 31
6D 70 34 31 20 20 20 08 66 72 65 65 20 2B BB 2122 6D 64 61 74 20 20 201D
C3 65 B8 40 F7 FF FE 1E 160 B3 20 29 2019 EE 20 29 BC 66 20 29 E6 2B 20
2A 10 07 20 2A 39 DD 20 2A 63 77 20 2A 8D 4F 20 2A B6 DA 20 2A E0 D8
20 2B 0A 69 20 2B 34 6B 20 2B 5D E7 20 2B 2021 B9 20 20 20 62 75 64 74 61
20 20 20 5A 6D 65 74 61 20 20 20 20 20 20 20 20 21 68 64 6C 72 20 20 20 20 20
20 20 20 6D 64 69 72 61 70 70 6C 20 20 20 20 20 20 20 20 20 20 20 2D 69
6C 73 74 20 20 20 25 A9 74 6F 6F 20 20 20 1D 64 61 74 61 20 20 20 01 20 20
20 20 4C 61 76 66 35 37 2E 31 39 2E 31 30 30 7B 65 6F 66 7D 7B 62 61 74 61
73 7D 4F 52 59 48

Tabel 4.3 Video setelah disisipkan text

Proses penyimpanan akan menyimpan pesan yang telah disembunyikan pada akhir file video tersebut.

b. 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.

5. Pengujian Aplikasi

Pada tahap pengujian penulis melakukan pengujian terhadap program yang telah dibuat. Pengujian yang dilakukan untuk mengetahui kemampuan aplikasi dalam menangani kondisi yang tidak diinginkan dengan tujuan untuk mempersiapkan banyak hal, selain itu untuk melakukan validasi dalam menerima dan jika dalam testing program menemui eror maka program bisa diperbaiki.

Pengujian aplikasi dilakukan dengan menggunakan metode pengujian yaitu pengujian *blackbox*

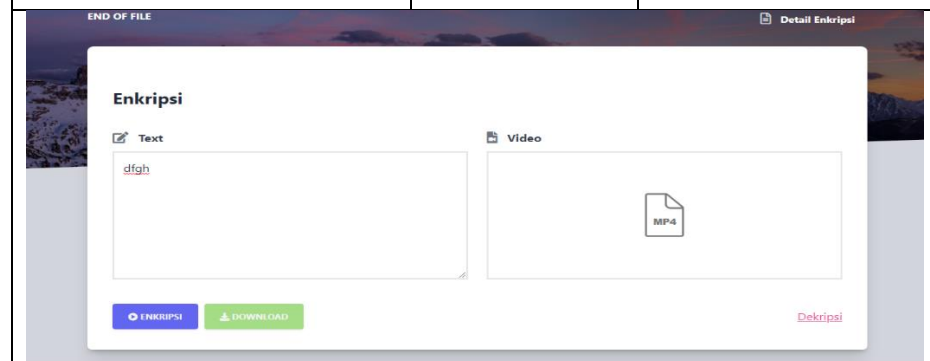
1. Pengujian Blackbox

Pengujian *blackbox* adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi *fungsionalitas*, khususnya pada input dan output aplikasi, apakah sudah sesuai dengan apa yang diharapkan atau belum.

a. *Blackbox* Enkripsi Video dan Pesan Berhasil

Tabel 4.4 Jika *Enkripsi* Pesan Berhasil

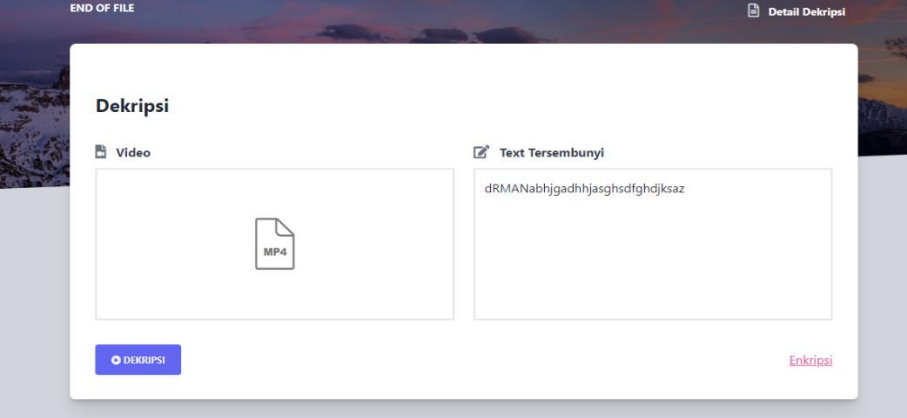
Test	Hasil	Kesimpulan
Jika pesan dan video berhasil di <i>enkripsi</i> pada menu <i>enkripsi</i>	✓	Berhasil, karena tidak ada muncul pesan validasi.



b. Blackbox Dekripsi Berhasil

Tabel 4.5 Jika *Dekripsi* Pesan Berhasil

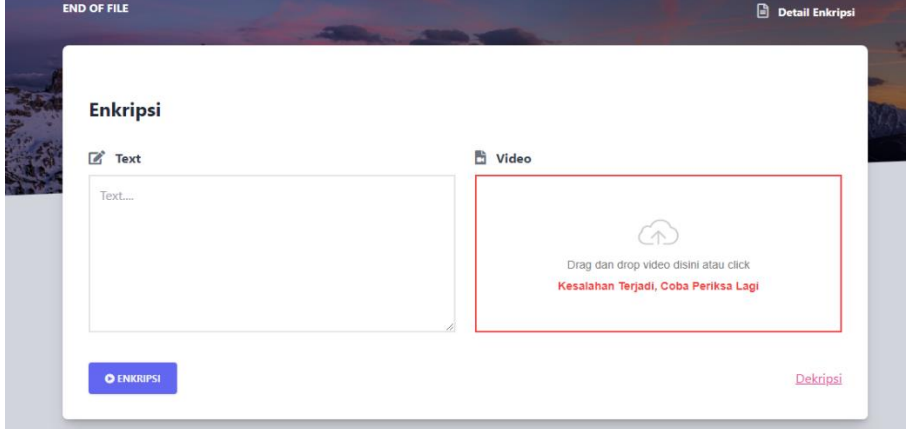
Test	Hasil	Kesimpulan
Jika video diinput pada menu <i>Dekripsi</i>	✓	Berhasil, karena tidak ada muncul pesan validasi.



c. Blackbox *Enkripsi* Bukan Video

Tabel 4.6 Jika diinput bukan video

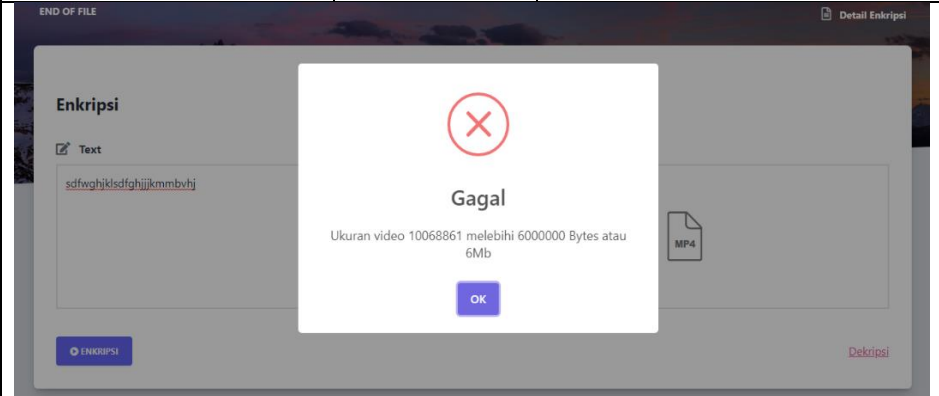
Test	Hasil	Kesimpulan
Jika yang diinput bukan video pada menu <i>Enkripsi</i>	✓	Berhasil, karena ada muncul pesan validasi.



d. Blackbox zise video besar ketika di Enkripsi

Tabel 4.7 Jika Zise Video Besar

Test	Hasil	Kesimpulan
Apabila video yang diinput terlalu besar atau lebih dari 6 MB	✓	Berhasil, karena ada muncul pesan validasi.



2. Hasil pengujian Enkripsi dan Dekripsi Video

Hasil pengujian proses *enkripsi* dan *dekripsi* adalah sebagai berikut :

Tabel 4. 8 Hasil pengujian proses *enkripsi* dan *dekripsi*

Video Sebelum Enkripsi	Durasi video	Pesan tersembunyi	Ukuran video Awal -Akhir	Berhasil / Gagal
Video alam 1 (*mp4)	17 detik	Strategi komunikasi humas PLTA bakaru dalam mengoptimalkan hubungan dengan masyarakat desa bakaru	3.726.656 bytes - 3.726.765 bytes	berhasil karena ukuran video tidak melebihi dari 6 MB
Video alam 2 (*mp4)	21 detik	alam ada,lebih dari sekedar membuat manusia hidup. alam juga tanda-tanda jalan	3.602.232 bytes - 3.602.346 bytes	berhasil karena ukuran video tidak melebihi dari 6 MB

Video Sebelum Enkripsi	Durasi video	Pesan tersembunyi	Ukuran video Awal -Akhir	Berhasil / Gagal
		pulang menuju kedamaian		
Video alam 3 (*mp4)	23 detik	alam telah hidup sejak jutaan ribu tahun lalu sehingga ia cukup bijaksana untuk mengajarkan kamu berbagai kehidupan melebihi orang yang kau anggap pintar	4.062.840 bytes – 4.063.005 bytes	berhasil karena ukuran video tidak melebihi dari 6 MB
Video alam 4 (*mp4)	23 detik	Penerapan teknologi dibidang pengamanan data dan informasi sudah banyak dilakukan. Ancaman terhadap keamanan informasi bisa terjadi ketika informasi yang dikirimkan tidak ditujukan kepada semua orang namun hanya kepada orang tertentu, terutama bila informasi yang diberikan bersifat rahasia.	4.517.926 bytes - 4.519.022 bytes	berhasil karena ukuran video tidak melebihi dari 6 MB
Video alam 5 (*mp4)	14 detik	Pengalaman memang membawamu pada kesuksesan, tapi keluar akan	33.208.111 bytes - 3.208.217 bytes	berhasil karena ukuran video tidak

Video Sebelum Enkripsi	Durasi video	Pesan tersembunyi	Ukuran video Awal -Akhir	Berhasil / Gagal
		mengajarkan soal makna kehidupan		
Video 6 (*mp4)	11 detik	Salah satu teknik steganografi adalah metode EOF. Data dimasukkan di akhir file menggunakan metode ini. Oleh karena itu, hal ini tidak akan berdampak pada kualitas data asli tempat pesan akan disisipkan.	22.866.367 bytes - 2.866.962 bytes	berhasil karena ukuran video tidak melebihi dari 6 MB
Video 07 (*Mp.4)	27 detik	Menjamin kerahasiaan dan keamanan data dan informasi sangat penting ketika berkomunikasi melalui internet. Dengan kemajuan teknologi saat ini.	5.972.744 bytes - 5.973.343 bytes	berhasil karena ukuran video tidak melebihi dari 6 MB
Video tari 8 (*Mp.4)	10 detik	perancangan dan analisis steganografi video dengan menyisipkan teks menggunakan metode end of file dan metode caesar chipertext	10.068.861 bytes	Gagal dikarenakan ukuran video melebihi dari 6 MB
	41 detik	File video terenkripsi akan	8.440.318 bytes	Gagal dikarenakan

Video Sebelum Enkripsi	Durasi video	Pesan tersembunyi	Ukuran video Awal -Akhir	Berhasil / Gagal
Video alam 9 (*Mp.4)		disimpan di penyimpanan yang tersedia setelah diunduh. Sementara itu, sistem terus diuji untuk mencapai hasil yang diinginkan.		ukuran video melebihi dari 6 MB
Video 10 (*Mp.4)	27 detik	Untuk selanjutnya diharapkan aplikasi ini dapat dikembangkan lebih lanjut tidak hanya untuk satu bahasa pemrograman saja tetapi juga berbagai bahasa pemrograman lainnya.	10.818.049 bytes	Gagal dikarenakan ukuran video melebihi dari 6 MB

BAB V

PENUTUP

A. Kesimpulan

Kesimpulan yang dapat diambil setelah melakukan penelitian dan hasil pengujian terhadap aplikasi, maka dapat ditarik kesimpulan sebagai berikut.

Menggunakan 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.

B. Saran

Adapun saran yang dapat diberikan untuk pengembangan pada penelitian ini :

1. Untuk selanjutnya diharapkan aplikasi ini dapat dikembangkan lebih lanjut tidak hanya untuk satu bahasa pemrograman saja tetapi juga berbagai bahasa pemrograman lainnya.
2. Untuk pengembangan selanjutnya video yang di enkripsi dan dekripsi tidak hanya untuk beberapa format ekstensi video saja tetapi bisa juga untuk semua format ekstensi video

DAFTAR PUSTAKA

- Andi Herul. (2019). Aplikasi Steganografi Menggunakan Metode LSB (*Least Significant Bit*)
- 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
- 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 *Spead 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*

