BAB I

PENDAHULUAN

A. Latar Belakang

Keamanan komputer menjadi bagian terpenting setelah penyebaran informai yang sangat pesat dan mudah diakses, dengan mudahnya informasi menyebar, berbagai metode keamanan ditingkatkan seperti mengamankan pesan dengan cara mengubah pesan menjadi sesuatu yang tidak dapat dimengerti oleh orang lain atau bisa disebut *kroptografi* dan juga dapat dilakukan dengan cara menyembunyikan data kedalam suatu media atau bisa disebut dengan *Steganografi*. Sebagai solusi masalah keamanan informasi adalah menerapkan ilmu *kriptografi* dan *steganografi*.

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Dony Ariyus, 2008:13). Adapun pesan yang akan dirahasiakan disebut dengan *plaintext*, sedangkan untuk pesan dari hasil penyandian disebut *ciphertext*. Pada proses penyandian *plaintext* menjadi *ciphertext* disebut *enkripsi* dan untuk proses membalikkan *ciphertext* menjadi *plaintext* disebut dekripsi.

Base64 merupakan salah satu algoritma untuk *encoding* dan *decoding* suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metode yang digunakan untuk melakukan *encoding* (penyandian) terhadap data binary.Untuk meningkatkan level keamanan

dan mencegah diketahuinya pesan *terenkripsi* yang disimpan ataupun dikirim dapat digunakan teknik penyembunyian pesan yaitu *steganografi* (Budi Darma 2019).

Steganografi merupakan suatu ilmu seni *dalam* menyembunyikan informasi dengan memasukkan informasi atau pesan tersebut ke dalam media lain, sehingga keberadaan informasi tersebut tidak diketahui oleh orang lain (Muh Ridwan,Yustikasari.2020). Media yang dapat dimanfaatkan untuk steganografi yaitu citra digital (image), teks, video, dan audio.

Discrete Cosine Transform atau DCT merupakan sebuah teknik untuk pengubahan sebuah sinyal kedalam suatu komponen *frekuensi* dasar. *Discrete Cosine Transform* mengolah sebuah gambar atau citra digital dari hasil penjumlahan suatu sinusoida dari magnitude dengan *frekuensi* yang berubah-ubah. Sifat dari metode DCT ialah merubah informasi gambar atau citra yang dikonsentrasikan pada beberapa koefisien DCT. Sistem penglihatan pada manusia tidak begitu *sensitive* dengan error-error yang ada pada frekuensi tinggi dibanding dengan yang ada pada *frekuensi* rendah. (Muh Ridwan ,Yustikasari,2020).

Secara matematis, citra merupakan fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi (Kusumanto dan Tompunu, 2011). Citra digital merupakan sebuah media *steganografi* yang sering digunakan untuk menyisipkan pesan. Penyisipan pesan ke dalam citra dapat menggunakan beberapa metode dalam *steganografi,* dan pada penelitian ini saya menggunakan *metode Distrate Cosines Transform (DCT)* dengan Bantuan metode *Least Significant Bit (LSB)*.

Berdasarkan uraian diatas, penulis bermaksud memadukan *kriptografi* algoritma *Base64* dengan metode *steganografi DCT* dengan bantuan lsb, proses yang dilakukan yaitu dengan menyembunyikan pesan yang telah ter*enkripsi* menggunakan algoritma *Base64* kedalam file image melalui teknik steganografi DCT dan LSB. Maka penulis melakukan penelitian yang berjudul "Aplikasi Perpaduan *Enkripsi* Algoritma *Base64* dengan Metode *Steganografi Discrete Cosine Transform*".

B. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka rumusan masalah yang dapat dibahas ialah:

1. Bagaimana memadukan algoritma enkripsi *Base64* dengan metode steganografi DCT menggunakan bahasa pemrograman PHP?
2. *Enkripsi dekripsi* pesan menggunakan algoritma *base64*
3. *Encode* dan *decode* teks ke gambar yang menggukan metode *disrete cosines transform (DCT)?*

C. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah dengan harapan penelitian terfokus dengan batasan-batasan yang dibuat. Adapun batasan dalam penelitian ini sebagai berikut :*Extensi* file yang digunakan hanya JPG.

1. File yang digunakan sebagai *plaintext* dalam proses *enkripsi* adalah berekstensi *.teks.

2. File yang digunakan pada teknik *steganografi* adalah file image berekstensi *.png.
3. File txt akan di enkripsi menggunakan *Base64* kemudian disembunyikan kedalam file image menggunakan teknik *DCT* dengan bantuan LSB.
4. Aplikasi yang dibangun menggunakan bahasa pemrograman *php*.

D. Tujuan

Sesuai dengan permasalahan yang telah diuraikan diatas, maka tujuan dari penelitian ini adalah untuk *enkripsi* teks dengan algoritma *base64* lalu di sisipkan dalam gambar dengan metode *steganografi* DCT.

E. Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Dapat menambah pengetahuan penulis dan mahasiswa lain tentang Aplikasi *kriftografi* dan *Steganografi*.
2. Memberikan sumbangan ilmiah dan sebagian referensi bagi peneliti lain mengenai *Kriftografi* dan *Steganografi.*
3. Diharapkan dapat memberi kemudahan dan manfaat untuk informasi secara akademis kepada pembaca tentang *Kriftografi* dan *Steganografi.*

F. Sistematika Penulisan

Dalam menyusun sistematika penulisan, penulis menguraikan ke dalam lima BAB yaitu :

1. Bab I Pendahuluan

Bab ini berisi uraian tentang apa, mengapa dan untuk apa suati topik diteliti. Bab ini terdiri dari latang belakang, rumusan masalah, tujuan penelitian, manfaat penelitian dan sistem penulisan.

2. Bab II Tinjauan Pustaka

Bab ini terdiri dari kajian teori dan kajian hasil penelitian terdahulu.

3. Bab III Metode Penelitian

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

4. Bab IV Hasil Penelitian dan Pembahasan

Pada bab ini dibahas hasil penelitian dan pembahasannya.

5. Bab V Penutup

Pada bab penutup mencakup kesimpulan dan saran. Keduanya dibahas secara terpisah.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Peneliti Terdahulu

1. Penelitian yang dilakukan oleh muhammad ridwan dan yusi yustikasari (2020) mengenai "Implementasi *steganografi text to image* dengan metode *distrete cosines transform*". Pada penelitian ini membahas tentang peneliti membuat suatu program implementasi *Steganografi* menggunakan *Discrete Cosine Transform*. Program ini akan menyediakan sebuah kemudahan dalam menyisipkan pesan yang berupa teks kedalam sebuah gambar atau citra digital dengan format *JPEG*. Steganografi menggunakan *Discrete Cosine Transform* ini menghasilkan gambar steganografi yang memiliki kualitas yang dominan sama dengan kualitas aslinya, hal ini dapat diketahui dengan melihat besarnya nilai rata-rata *Peak Signal to Noise Ratio* atau PSNR dari beberapa gambar yang berbeda.
2. Penelitian yang dilakukan oleh Muhamad Fitra Syawal dan Deddy Chandra Fikriansyah (2016) mengenai "Implementasi Teknik *Steganografi* Menggunakan Algoritma *Vigenere Cipher Dan Metode LSB*". Pada penelitian ini menbahas peroses kombinasi kriftigrafi dan steganografi pesan teks *(plaintext)* yang kan disisipkan kedalam citra digital terlebih dahulu dienkripsi dengan menggunkan metode *vigenere chiver,*selanjutnya

hasil enkripsi *(ciphertext)* disispkan dengan menggunakan *LSB* hasil dari penyisipan berupa citra hasil (*stegano image).*

3. Penelitian yang dilakukan oleh Endang Ratnawati Djuwitaningrum dan , Melisa Apriyani (2016). mengenai "Teknik *Steganografi* Pesan Teks Menggunakan *Metode Least Significant Bit (LSB)*". Pada penelitian ini membahas tentang peneliti menggabungkan *kriptografi* dengan *Steganografi.* Pesan yang akan disisipkan terlebih dahulu dienkripsi. Selanjutnya pesan yang sudah terenkripsi disisipkan ke dalam cover image menggunakan teknik *LSB.*

4. Sama hal nya dengan penelitian yang saya lakukan pesan yang akan disisispkan terlebih dahulu di *enkripsi* dengan algoritma *base64* selanjutnya pesan terenkripsi disisipkan kedalam cover image png dengan metode *steganografi DCT.*

B. Aplikasi

Menurut Kamus Besar Bahasa Indonesia (KBBI), aplikasi adalah penerapan dari rancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Dari definisi di atas dapat disimpulkan bahwa aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas tertentu untuk pengguna.

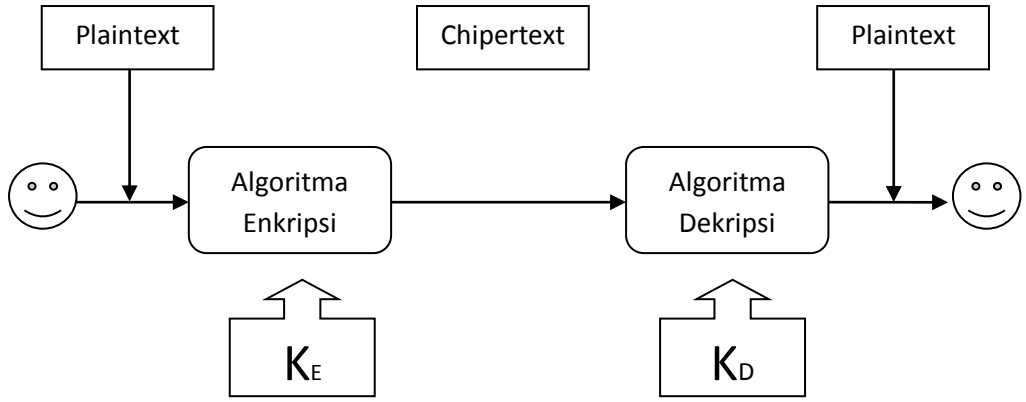
C. *Kriftografi*

Kriptografi berasal dari bahasa yunani, crypto dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya,*kriptografi* adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. (Dony Ariyus,2008).

Kriptografi adalah ilmu dan seni dalam mengamankan pesan dengan cara mengubah pesan menjadi sesuatu yang tidak dapat dimerngerti oleh orang lain dengan teknik-teknik dan metode-metode tertentu. *Kriptografi* tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya.

D. Sistem *Kriptografi*

Crypto system digunakan untuk menjamin privasi dan autentik data dalam sistem komputer komunikasi. *Message* yang tidak diproteksi disebut *plaintext*. Proses tersebut yang mana *plaintex*t dibentuk dalam *ciphertext* dari suatu bentuk yang tak dapat dipahami yang disebut *enkripsi* atau *enchiperment*. Sebuah algoritma *dechipering* digunakan untuk *dekripsi* atau *decipherment* agar mengembalikan *plaintext* aslinya. Dalam *cryptosystem*,sekumpulan parameter yang memilih sebuah transformasi *chipering* khusus yang disebut sekumpulan key (Didik, 2009). *Enkripsi* dan *dekripsi* dikontrol oleh sebuah key atau beberapa key.



Gambar 2.1. Cryptosystem

Secara umum operasi enchipering dan dechipering dijelaskan sebagai berikut :

Y = EKE (X) (enkripsi) (2.1)

X = EKD (Y) (enkripsi) (2.2)

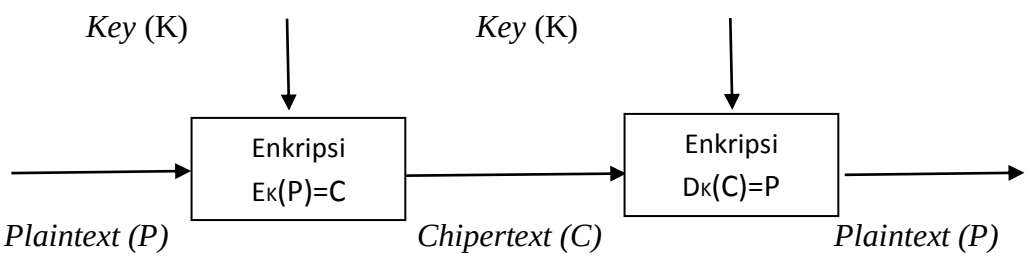
dimana :

X =plaintext

Y=ciphertext

KE=key enkripsi

KD=key dekripsi

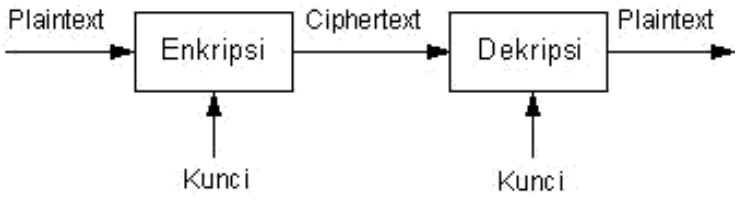


Gambar 2. 1 Ilustrasi *enkripsi* dan *dekripsi*

E. Algoritma *Kriptografi*

Algoritma kriptografi terdiri dari tiga aspek dasar (Dony Ariyus, 2008) yaitu:

1. *Enkripsi* merupakan hal yang sangat penting dalam *kriptografi*, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *plaintext*, yang di ubah menjadi kode-kode yang tidak dimengerti atau disebut *ciphertext*.
2. *Dekripsi* merupakan kebalikan dari proses *enkripsi*. Pesan yang *enkripsi* dikembalikan ke bentuk asalnya atau dengan kata lain proses membalikkan *ciphertext* menjadi *plaintext.*
3. Kunci yang dimaksud disini adalah kunci yang dipakai untuk melakukan *enkripsi* dan *dekripsi*. Kunci terbagi menjadi dua bagian, kunci rahasia *(private key)* dan kunci umum *(public key).* Ilustrasi untuk menggambarkan kondisi *kriptografi* pada gambar di bawah:

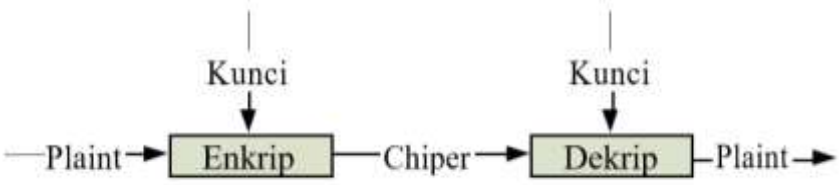


Gambar 2. 3 Proses *enkripsi* dan *dekripsi*

Berdasarkan kunci yang digunakan dalam proses *enkripsi* dan *dekripsi*, algoritma kunci *kriptografi* dapat dibedakan menjadi 2 bagian, yaitu algoritma *kriptografi* kunci *simetri* dan algoritma *kriptografi* kunci *asimetri* (Dony Ariyus, 2008).

1. Algoritma *Kriptografi* Kunci Simetri

Algortima ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan *enkripsi* dan *dekripsi*. Algoritma ini sudah ada sejak lebih dan 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsikan pesan yang akan dikirim. Keamaan dari pesan yang menggunakan algoritma ini tergantung kunci. Jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan *enkripsi* dan *dekripsi* terhadap pesan. Algortima yang memakai kunci simetri diantaranya adalah *Base64.* Proses *enkripsi-dekripsi* algoritma *kriptografi* kunci *simetri* dapat dilihat pada gambar 2.4 berikut :



Gambar 2. 4 Proses *enkripsi-dekripsi* kunci simetris

Algoritma kriptografi simetris dibagi menjadi dua kategori yaitu

a. Algoritma aliran *(Stream Chipers)*

b. Algoritma blok *(Blok Chipers)*

F. Base64

Base64 merupakan istilah umum untuk sejumlah skema pengkodean serupa yang digunakan untuk melakukan proses *encoding* dan *decoding* terhadap suatu data ke dalam format *ASCII*,dan menerjemahkannya ke dalam representasi

basis 64 atau biasa dikatakan sebagai salah satu metode yang digunakan untuk melakukan proses *encoding* (penyadian) terhadap data biner. Algoritma ini menggunakan kode ASCII dan kode index *Base64* dalam melakukan proses *enkripsi* ataupun *dekripsinya* (Davit Herianto,2013).

Skema *enkripsi Base64* biasanya digunakan ketika diperlukan sandi terhadap data biner yang didesain untuk menangani data berbentuk teks, hal ini ditujukan untuk menjaga data selama pengiriman ke suatu server. Karakter yang dihasilkan pada transformasi *Base64* terdiri dari A..Z, a..z, dan 0..9, serta ditambahkan dengan dua karakter terakhir yang bersimbol + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data binary atau istilahnya disebut sebagai pengisi pas. Karakter simbol yang dihasilkan akan tergantung proses algoritma yang berjalan.(Ros minarni 2019).

(Budi Darma 2019) Istilah *Base6*4 berasal dari konten pengkodean *MIME (Multipurpose Internet Mail Extensions)* pada *RFC (Request For Comments)* 2045 tahun 1996. *Base64* biasanya digunakan ketika ada kebutuhan untuk menyandikan data biner yang perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani data tekstual. Hal ini untuk memastikan bahwa data tetap utuh tanpa perubahan selama pengiriman. *Base64* digunakan umum dalam beberapa aplikasi termasuk email melalui *MIME,* dan penyimpanan data yang kompleks dalam *XML(Extensible Markup Language).* Transformasi *Base64* banyak digunakan di dunia Internet sebagai media data format untuk mengirimkan data. Dikarenakan hasil dari transformasi *Base64* berupa *plaintext*, maka nilai ini akan jauh lebih. mudah dikirim, dibandingkan format data berupa binary.

Tabel 2.1 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	□

Table 2. 2 Index Base64

Value	Char		Value	Char		Value	Char		Value	Char
0	A		16	Q		32	g		48	w
1	B		17	R		33	h		49	x
2	C		18	S		34	i		50	y
3	D		19	T		35	j		51	z
4	E		20	U		36	k		52	0
5	F		21	V		37	l		53	1
6	G		22	W		38	m		54	2
7	H		23	X		39	n		55	3
8	I		24	Y		40	o		56	4
9	J		25	Z		41	p		57	5
10	K		26	a		42	q		58	6
11	L		27	b		43	r		59	7
12	M		28	c		44	s		60	8
13	N		29	d		45	t		61	9
14	O		30	e		46	u		62	+
15	P		31	f		47	v		63	/
Pad	=									

G. *Stegnografi*

Kata *steganografi (steganography)* berasal dari bahasa Yunani yaitu steganos yang berarti tersembunyi atau terselubung dan graphia yang artinya menulis, sehingga arti *steganografi* adalah "menulis (tulisan) terselubung" (Darmayanti, 2016). *Steganografi* adalah ilmu menyembunyikan teks pada media lain yang telah ada sedemikian sehingga teks yang tersembunyi menyatu dengan media itu. Media tempat penyembunyian pesan dapat berupa media teks, image, audio atau video (Sadikin, 2012:10).



Gambar 2. 5 Proses *Embedding* dan *Ekstraksi*

Berdasarkan gambar proses *steganografi* menggunakan media citra diawali dengan menginput pesan dan *cover image* (citra yang digunakan sebagai media penyisipan). Selanjutnya dilakukan proses *embedding,* sehingga diperoleh stegoimage citra yang telah disisipkan pesan). *Stego image* inilah yang akan dikirim ke penerima pesan. Penerima pesan melakukan *extraction* (proses pengeluaran pesan pada citra). Setelah melakukan proses extraction, maka pesan yang dikirim dapat dibaca.

H. Algoritma

Menurut Donald E.Knuth (1973:4), dalam pengertian modern mempunyai kemiripan dengan istilah resep, proses, metode, teknik, prosedur, rutin. Algoritma adalah sekumpulan aturan-aturan berhingga yang memberikan sederetan operasi-operasi untuk menyelesaikan suatu jenis masalah yang khusus. Menurut Rinaldi Munir (2006), Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Berdasarkan dua pengertian algoritma di atas, dapat disimpulkan bahwa algoritma merupakan suatu istilah yang luas, yang tidak hanya berkaitan dengan dunia komputer.

Algoritma adalah independen terhadap bahasa pemrograman tertentu, artinya algoritma yang telah dibuat tidak boleh hanya dapat diterapkan pada bahasa pemrograman tertentu. Penulisan algoritma tidak tertikat pada suatu aturan

tertentu, tetapi harus jelas maksudnya untukt tiap langkah algoritmanya. Namun pada dasarnya algoritma dibagi menjadi beberapa bagian macam berdasarkan format penulisannya yaitu :

1. *Deskriptif*

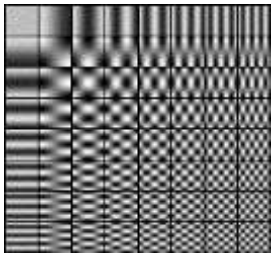
Algoritma *deskriptif* merupakan algoritma yang ditulis menggunakan bahasa manusia sehari-hari (misalnya dalam bahasa Indonesia atau bukan dengan bahasa Inggris) dan dalam bentuk kalimat. Setiap langkah algoritmanya diterangkan dalam satu atau beberapa kalimat.

2. Flowchart (Diagram Alir)

Tidak hanya dalam bentuk tulisan, algoritma juga bisa ditulis dalam bentuk diagram-diagram menggunakan anak panah sebagai penanda untuk urutan langkah anda pada suatu algoritma. Algoritma yang ditulis dengan simbol-simbol demikian yang digunakan *flowchart.*

I. *Discrete Cosine Transform*

Discrete Cosine Transform atau bisa juga disebut teknik kompresi adalah sebuah teknik untuk pengubahan sebuah sinyal kedalam suatu komponen *frekuensi* dasar. *Discrete Cosine Transform* mengolah sebuah gambar atau citra digital dari hasil penjumlahan suatu sinusoida dari magnitude dengan *frekuensi* yang berubah-ubah. Sifat dari metode DCT ialah merubah informasi gambar atau citra yang dikonsentrasikan pada beberapa koefisien DCT. Sistem penglihatan pada manusia tidak begitu *sensitive* dengan error-error yang ada pada frekuensi tinggi dibanding dengan yang ada pada *frekuensi* rendah. (Yustikasari,2020).



Gambar 2.6 *Frekuensi* Pada *Block Discrete Cosine Transform* DCT

DCT pada kompresi png menerima sebuah masukan berupa matriks-matriks gambar yang berukuran 4x4, kemudian mengubahnya menjadi sebuah matriks frekuensi berukuran sama. Sedangkan kebalikan dari proses ini aialah IDCT, yaitu proses tersebut akan mengembalikan sebuah koefisien pada suatu matriks frekuensi menjadi sebuah matriks gambar.

Terdapat beberapa jenis koefisien pada sebuah matriks frekuensi yang terdapat pada penelitian, yaitu koefisien DC dan AC. Koefisien DC merupakan nilai pada frekuensi 0. Jumlah koefisien ini hanya satu, yang terletak pada sudut kiri atas matriks frekuensi. Sedangkan 63 koefisien lainnya merupakan koefisien AC yang frekuensinya lebih besar dari 0, semakin ke kanan maka menunjuk pada frekuensi horizontal yang semakin tinggi, semakin ke bawah maka akan menunjuk pada frekuensi vertical yang semakin tinggi.

Alur Algoritma menggunakan DCT pada Kompresi png:

1. Gambar asli akan dibagi menjadi blok-blok yaitu menjadi 4 x 4.
2. Nilai-nilai pada pixel dari berbagai gambar hitam dan putih adalah kisaran antara 0-255 tapi DCT ini dirancang untuk bekerja pada nilai-nilai piksel mulai dari kisaran -128 sampai 127.
3. Persamaan digunakan untuk menghitung nilai DCT ada matriks.

4. DCT ini diterapkan untuk setiap blok-blok dengan mengalikan blok yang diubah dengan DCT matriks disebelah kiri dan akan ditranspos dari matriks DCT disebelah kanannya.
5. Setiap blok kemudian akan dikompresi melalui kuantisasi. Matriks terkuantisasi kemudian akan dikodekan entropi.

Berikut adalah nilai ketentuan spatial domain frekuansi untuk matrik 4x4.

52	55	61	66
63	59	66	90
62	59	68	113
63	58	71	122

Gambar 2.7 *spatial domain frekuensi*

Menurut Trusha patel (2020) *Discrete Cosine Transform* dua dimensi dapat diperoleh dengan menggunakan rumus berikut:

$$F = C f C^T$$

$$C(u,v) = \begin{cases} \sqrt{\frac{1}{N}} & u=0,\ 0 \le v \le N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2v+1)\pi u}{2N}\right) & 1 \le u \le N-1,\ 0 \le v \le N-1 \end{cases}$$

Gambar 2.7 Rumus Dct

Keterangan :

F : Frekuansi Domain

C^T : Cosines transform

f : Spatial Domain

N : Jumlah matrixs

C : Cosines

1. Jenis metode kompresi

a. Loseless Data Compression

Loseless Data Compression adalah teknik kompresi data dimana data kompresi dapat didekompres kembali. Selain itu hasil kompresi data yang dihasilkan sama persis seperti data asli sebelum dilakukan kompresi. Contoh dari *Loseless Data Compression* adalah *ZIP, RAR dan 7-ZIP*. Pemakaian *Loseless Data Compression* biasanya untuk akurasi data yang sangat penting seperti data teks, data program dan Image *(PNG, GIF).*

b. Lossy Data Compression

Lossy Data Compression adalah teknik pemampatan data yang hasilnya tidak sama dengan data aslinya, namun tidak menjadikannya masalah. Teknik *Lossy Data Compression* ini contohnya adalah *MP3, Streaming Media, JPEG, MPEG dan WMA.* Penggunaan teknik *Lossy Data Compression* data ini dinilai lebih efisien dibandingkan dengan *Loseless Data Compression*. Alasannya yaitu teknik *Lossy Data Compression* hanya membuang bagian-bagian data yang kurang penting. Oleh sebab itu data yang dihasilkan masih bisa digunakan meskipun beberapa bagian dari data aslinya ada yang dihilangkan.

J. Least Significant Bit

(Adiria.2010) Metode *Steganografi Least Significant Bit (LSB)* Penyembunyian data dilakukan dengan mengganti bit-bit data yang tidak terlalu

berpengaruh di dalam segmen citra dengan bit-bit data rahasia (Jonathan Cummins, 2004), Pada susunan bit di dalam sebuah byte (1 *byte* = 8 bit), ada bit yang paling berarti *(most significant bit atau MSB)* dan bit yang paling kurang berarti *(least significant bit atau LSB).* Perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut menyatakan warna merah, maka perubahan satu bit *LSB* tidak mengubah warna merah tersebut secara berarti. Lagi pula, mata manusia tidak dapat membedakan perubahan yang kecil.

Hal ini penyisipan *Least Significant Bit* (LSB) merupakan pendekatan yang umum untuk menanamkan informasi dalam media citra. *Least Significant Bit* (dengan kata lain bit ke-8) sebagian atau seluruh dari *byte* dalam sebuah gambar diubah menjadi sebuat bit dari pesan rahasia.

Pada citra 8 bit yang berukuran 256 x 256 *pixel* terdapat 65.553 *pixel*, setiap *pixel* berukuran 1 *byte* sehingga hanya dapat menyisipkan 1 *bit* pada setiap *pixel*. Pada citra 24 *bit* yang berukuran 256 x 256 *pixel*, satu *pixel*berukuran 3 *byte* (atau 1 *byte* untuk setiap komponen R,G, B),sehingga bisa menyisipkan pesan sebanyak 65.553 x 3 bit = 196.608 bit atau 196.608 / 8 = 24.576 *byte*.

Pada uji coba ini digunakan media citra *digitaltrue color*24 *bit* dengan model warna RGB. Pada citra digital nantinya 3 bit yang dapat disisipkan dalam 1 dikarenakan dalam 1 *pixel* warna tersusun dari 3 komponen warna yaitu *Red, Green*, dan *Blue* yang masing-masing disusun oleh 8 digit bilangan

biner dari rentang nilai 0 sampai 255 dalam desimal atau 00000000 sampai 11111111 dalam representasi biner.

K. PHP (Hypertext Preprocessor)

Supono dan Putratama (2016) mengemukakan bahwa ”*PHP (PHP Hypertext Preprocessor)* adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis server-side yang dapat ditambahkan ke dalam *HTML*”. Sedangkan, menurut Solichin (2016) mengemukakan bahwa “PHP merupakan salah satu bahasa pemrograman berbasis web yang ditulis oleh dan untuk pengembang web*”. PHP* merupakan bahasa *(script)* pemrograman yang sering digunakan pada sisi server sebuah web.

Kumpulan kutipan diatas menerangkan bahwa *hypertext preprocessor (PHP)* merupakan bahasa pemrograman untuk membuat atau mengembangkan aplikasi berbasis web dan bersifat *open soure* dan ditanamkan ke dalam *script HTML.*

L. *HTML (Hypertext Markup Language)*

HTML (Hypertext Markup Language) merupakan kerangka dasar untuk tampilan pada browser, menggunakan bahasa markah sebagai dasar pengkodean. Hingga pada saat ini *HTML* menggunkan versi ke 5.

Menurut Prasetio (2010) mengemukakan bahwa *HTML* merupakan “bahasa pemrograman yang digunakan untuk mendesain sebuah halaman web”. Sedangkan menurut Solichin (2016) mengemukakan bahwa “*HTML* merupakan bahasa pemrograman *web* yang memberitahukan peramban web *(web*

browser) bagaimana menyusun dan menyajikan konten di halaman web". **HTML** adalah bahasa markup untuk menyebarkan informasi pada web (Simarmata,2010). Berdasarkan teori dari para ahli di atas, maka *hyptertext* markup language *(HTML)* merupakan bahasa pemrograman yang dikenal oleh browser untuk menampilkan informasi lebih menarik di halaman web melalui web browser.

M. *XAMPP*

Menurut Yogi wicaksono (2008:7) "*XAMPP* adalah sebuah software yang berfungsi untuk menjalankan website berbasis *PHP* dan menggunakan pengolah data *MySQL* dikomuter local". *XAMPP* berperan sebagai server app pada komputer anda. *XAMPP* juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membuat anda melakukan *preview* sehingga dapat memodifikasi website tanpa harus online atau terakses dengan internet. *XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program *Apache HTTP* Server, *MySQL* database, dan penerjamah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache, MySQL, PHP* dan *Perl.*

N. *Google Chrome*

Google Chrome merupakan salah satu *browser* yang dikembangkan oleh google, berfungsi untuk melakukan browsing. *Google chrome* sama dengan kebanyakan browser modern lainnya, menggunakan dan membaca *HTML, CSS*

dan *Javascript* sebagai dasar penggunakan aplikasi.

Menurut Jubliee (2009) *Google Chrome* "Sebagai *browser* baru mempunyai fasilitas yang lumayan bagus sehingga mampu menarik perhatian pecinta dunia maya dari seluruh penjuru dunia ".

Google Chrome merupakan mesin pencarian mampu melakukan penelusuran dalam waktu kurang dari beberapa detik dengan perangkat lunak yang telah diinstal ke dalam Sistem Operasi windows untuk memberikan pengguna aksesoris pendukung seperti mediator layanan browser, file manager, *downloader* dan lain-lain. Kombinasi tampilan teknologi canggih membuat Google Chrome menjadi sangat diminati pengguna sebagai browser canggih yang dimanfaat saat ini. Sebagai salah satu layanan software yang memungkinkan pengguna website menelusuri informasi, media video dan audio, serta data teknis *Google Chrome* tersedia dan sangat mendukung untuk semua Operasi Sistem Dektop hingga pengguna smartphone seperti Android dan *Apple* agar *broswer* menjadi terkendali untuk diterima, ditelususri, disimpan hingga digunakan sebaik-baiknya dalam dunia maya. (McFedries, 2010).

O. Web Server

Menurut Supardi (2010) *web server* adalah "Perangkat lunak yang mengelola (mengatur) permintaan *user* dari *browser* dan hasilnya dikembalikan lagi kepada browser".

Web server memiliki peran sebagai klien yang mampu bertindak mentransfer data dalam satu koneksi pada jaringan komputer. Dari transfer data tersebut, web server menampilkan informasi dasri suatu situs yang pengguna

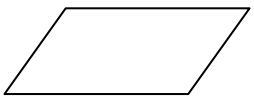
gunakan pada *web server*. Dengan adanya *web server*, internet dapat berfungsi menjalankan personal aplikasi web server seperti *Personal Web Server, IIS (InternetInformation Service) dan Apache* untuk mengeksekusi bahasa web programming seperti *HTML, PHP, Jquery* dan lain-lain. (Ramadani, 2010).

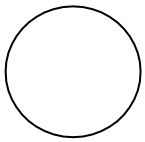
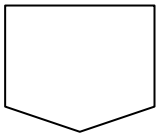
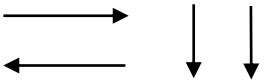
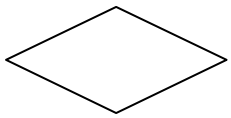
Localhost merupakan *web server* yang diperuntukkan berjalan pada satu perangkat, dan akan diakses secara internal oleh pengguna pada perangkat tersebut.

P. Bagan Alir Program

Bagan alir program *(flowchart)* merupakan bagan yang menjelaskan rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir sistem (Jugiyanto, 2005). Bagan alir program dibuat dengan menggunakan symbol-symbol sebagai berikut :

Tabel 2.3. Simbol *Flowchart*

Simbol	Keterangan
	Simbol *input/output* digunakan untuk mewakili data *input/output*.
	Simbol proses, digunakan untuk mewakili suatu proses.
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.

	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus dihalaman yang masih sama.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus dihalaman yang berbeda.
	Simbol garis alir, menunjukkan arus dari proses
	Simbol keputusan, digunakan unuk suatu penyelesaian kondisi didalam program.

Q. *UML (Unified Modelling Language)*

Menurut Nugroho (2010), *UML (Unified Modelling Language)* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan *(modelling)* sesungguhnya digunakan untuk penyederhanaan permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML *(Unified Modeling Language)* adalah sekumpulan diagram yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek UML dapat digunakan untuk mempermudah pengembangan aplikasi yang berkelanjutan. UML dapat dikatakan juga sebagai perkembangan, bahasa pemodelan di bidang rekayasa perangkat lunak yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sebuah sistem. UML terdiri dari banyaknya elemen-elemen grafis yang digabungkan dalam bentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model.

1. Sesuatu *(things)*

Ada empat *things* dalam *Unified Modeling Langueage (UML)* :

a. *Structural things*

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

b. *Behavorial things*

Bagian dinamis biasanya merupakan kata kerja dari model *UML* yang mencerminkan perilaku sepanjang waktu.

c. *Grouping things*

Bagian pengorganisasian dalam *UML*. Penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket

ini kemudian dapat didekomposisikan lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem-subsistem.

d. An notational thing

Merupakan bagian yang memperjelas model *UML*. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

2. **Relasi *(relationship)***

Ada empat relationship (hubungan) dalam *Unified Modeling Language (UML) :*

a. **Keterangan *(dependency)***

Adalah Hubungan dimana perubahan yang terjadi pada suatu elemen independen akan mempengaruhi elemen yang bergantung padanya.

b. Asosiasi

Adalah apa dan bagaimana yang menghubungkan antara objek satu dengan lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungkan suatu objek dengan bagian-bagiannya.

3. Diagram

Unified Modeling Language (UML) menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

Tujuan atau fungsi dari penggunaan UML antara lain sebagai berikut:

a. Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
b. Dapat menyatukan praktik-praktik terbaik yang ada dalam pemodelan.

c. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemodelan visual yang efspresif untuk menembangkan sistem dan untuk saling menukar model secara mudah.
d. Dapat berguna sebagai blue print sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
e. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (software) saja.

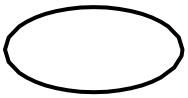
Menurut Herlawati (2011), beberapa literatur menyebutkan bahwa *UML* menyediakan sembilan jenis diagram. Adapula diagram yang sering digunakan pada penelitian dan perancangan sistem adalah *Use Case Diagram, Activiti Diagram, Sequence Diagram*.

1) Use Case Diagram

Menurut Murad (2013), Use Case Diagram adalah diagram yang bersifat status yang memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini memiliki 2 fungsi, yaitu mendefinisikan fitur apa yang harus disediakan oleh sistem dan menyatakan sifat sistem dari sudut pandang user.

Use Case mendeskripsikan interaksi tipikal antara para pengguna sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Dua hal penting dalam use case adalah skenario dan aktor. Skenario adalah rangkaian langkah-langkah yang menjabarkan sebuah interaksi antara seorang pengguna dengan sebuah sistem.

Tabel 2.4. Simbol *Usecase diagram*

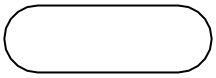
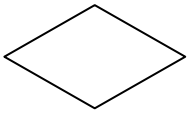
Simbol	Nama	Keterangan
	Aktor	Sesesorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun.
	*Use Case*	Menggambarkan bagaimana seseorang menggunakan sistem.
	Relasi Asosiasi	Relasi yang dipakai untuk menunjukkan hubungan antara aktor dan use case
<<include>> 	*Relasi Include*	Memungkinkan suatu *use case* menggunakan fungsionalitas yang disediakan oleh *use case* lain
<<extends>> 	*Relasi Extend*	Memungkinkan suatu *use case* secara optional menggunakan fungsionalitas yang disediakan oleh *use case* lainnya

2) Activity Diagram

Activity Diagram digunakan untuk menampilkan rangkaian kegiatan, menunjukkan alur kerja dari suatu titik awal ke titik akhir keputusan, merinci banyak jalur yang ada dalam perkembangan peristiwa yang terkandung dalam kegiatan (Adi Nugroho, 2010).

Diagram aktivitas memiliki komponen dengan bentuk tertentu, di hubungkan dengan tanda panah. Panah mengarahkan urutan aktivitas yang terjadi dari awal sampai akhir.

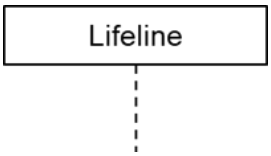
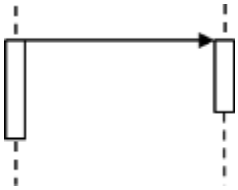
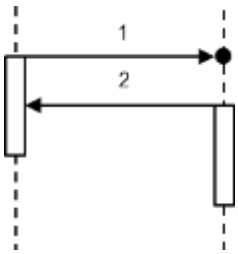
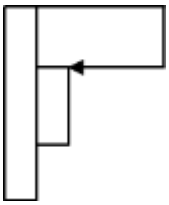
Tabel 2.5. Simbol *Activity Diagram*

Simbol	Nama	Ketenrangan
	*Start state*	Titik awal atau permulaan.
	*End state*	Titik akhir atau akhir dari aktivitas.
	*Activity*	*Activity* atau aktivitas yang dilakukan oleh aktor.
	*Decision*	Memungkinkan suatu *use case* menggunakan fungsionalitas yang dilakukan oleh *use case* lainnya.
	Interaction	Alur

3) Sequence Diagram

Sequence Diagram adalah diagram interaksi yang disusun berdasarkan urutan waktu. Setiap diagram, sekuensial mempresentasikan satu flow dari beberapa di dalam use case (Adi Nugroho, 2010).

Tabel 2. 6. *Sequence Diagram*

Simbol	Nama	Keterangan
	*Lifeline*	Mengambarkan sebuah objek dalam sebuah sistem atau salah satu komponennya.
	*Create Message*	Pembuatan sebuah *message* sederhana antar elemen dan juga mengindikasikan komunikasi antara objek.
	*Synconous Message*	*Message* ini mengaktifkan sebuah proses dan sampai selesai, baru bisa mengirimkan sebuah *message* baru.
	*Message to self*	Suatu hasil kembalian sebuah operasi dan berjalan pada objek itu sendiri.

R. Kerangka Pikir

Minimnya tingkat keamanan informasi dari sebuah pesan menyebabkan seringkali pesan yang ingin disampaikan mudah diketahui orang lain selain penerimanya sendiri.



Membuat sebuah aplikasi *encode* dan *decode* yang dapat digunakan untuk menyembunyikan informasi menggunakan image dengan menerapkan *kriptografi* dan *steganografi.*



Memadukan algoritma *enkripsi* dan *Base64* dengan metode *steganografi* DCT *(Distrete Cosines Transform)* dengan bantuan metode LSB.



Hasil dari penggabungan antara teknik *kriptografi* dan teknik *steganografi* ini, gambar berformat png yang sudah disisipkan pesan.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang dilakukan merupakan penelitian eksperimental, yaitu penelitian yang pengumpulan datanya melalui pencatatan secara langsung dari hasil percobaan yang dilakukan.

Tabel 3. 1. Jadwal pelaksanaan penelitian

No	Uraian Kegiatan	Bulan 2021	
		Agustus	September
1	*Studi Literatur*		
2	Analisis data		
3	Perancangan aplikasi		
4	Pembuatan aplikasi		
5	Pengujian		

B. Lokasi dan Waktu

Lokasi penelitian dilaksanakan di Universitas Muhammadiyah Parepare dan waktu penelitian dilakukan selama 2 (dua) bulan dimulai pada bulan agustus sampai dengan bulan septamber 2021.

C. Metode Pengumpulan Data

Pada metode pengumpulan data untuk mendapatkan data yang cukup akurat, maka digunakan beberapa metode di bawah 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 *Kriftografi* dan *Steganografi* .

2. Eksperimen

Setelah mendapatkan data secara studi pustaka, proses penelitian akan dilakukan eksperimen atau percobaan. Dalam eksperimen ini pengumpulan data dapat diambil secara langsung, sehingga akan lebih mendalami dalam melakukan penelitian. Penelitian juga meminimalkan kesalahan dalam penelitian ini.

D. Alat dan Bahan Penelitian

Penelitian ini digunakan beberapa alat, yang terdiri dari :

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 mebangun 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

E. Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah sebagai berikut :

1. Pengumpulan data

Pada tahap ini dilakukan pengumpulan data-data yang menunjang keberhasilan dari penelitian, seperti melakukan studi literatur tentang aplikasi perpaduan *enkrifsi* algoritma *base64* dengan metode steganografi *DCT*.

2. Analisis data

Pada tahap ini dilakukan analisis terhadap data yaitu *file type* txt yang akan di sisipkan dalam gambar.

3. Perancangan aplikasi

Pada tahap ini yang dilakukan perancangan *flowchart, UML*, dan desain dari aplikasi.

4. Pembuatan aplikasi

Aplikasi dibangun menggunakan kombinasi dari algoritma *enkripsi base64* dengan metode *steganografi dct* dan menggunakan bahasa pemrograman PHP.

5. Pengujian aplikasi

Aplikasi yang telah dibangun akan diuji keberhailannya dalam melakukan penyisipan tekt kedalam gambar .

6. Implementasi

Implementasi akan dilakukan setelah pengujian terhadap aplikasi berhasil dilakukan.

F. Metode Pengujian

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, query, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk

menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

Coba *white box testing* merupakan metode perancangan *test case* yang mengunakan struktural untuk mendapatkan *test case*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di tes dengan kondisi dan perulangan secara fisik. Black box testing merupakan pengujian yang bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya program, Jadi semua proses yang ada pada aplikasi akan diuji dengan metode *black box* apakah perangkat lunak dapat beroprasional, bahwa input diterima dengan baik dan uotput di hasilkan dengan tepat.

Contoh pengujian *white box* testing ini merupakan peringatan ketika user menginputkan password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada user mengenai kesalahan yang di lakukan.

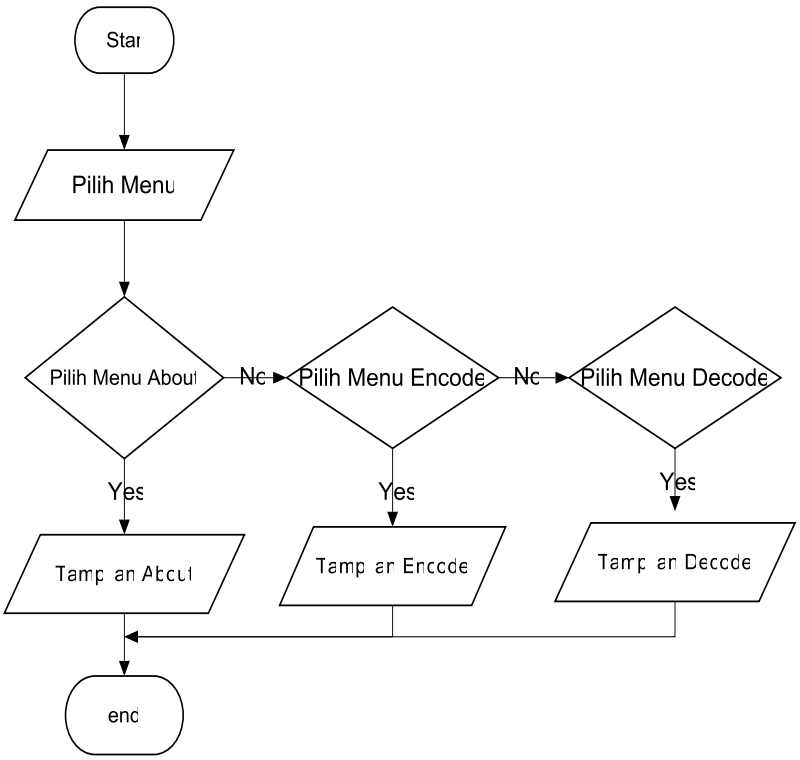
2. *Black box*

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika user belum menginput data yang harusnya di input sesuai ketentuan sistem yang di jalankan dan sistem memberikan informasi kepada userkarena data yang ingin diproses belum lengkap atau tidak memenuhi

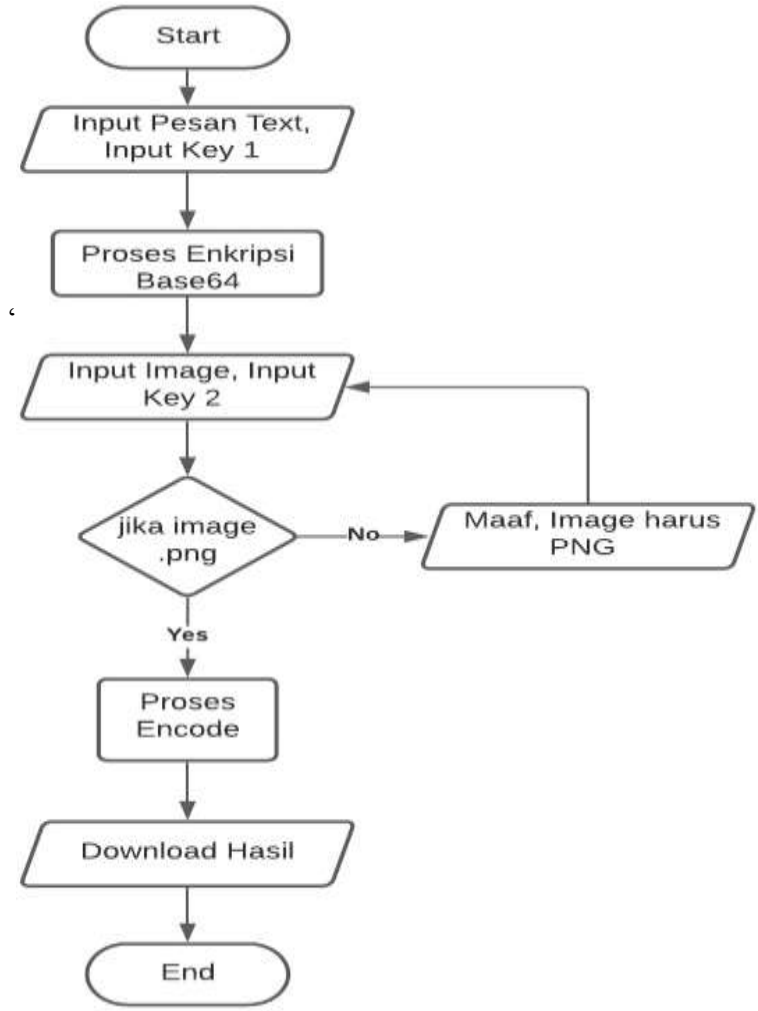
ketentuan untuk proses selanjutnya. Pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya *(basis path)* atau proses *looping* (pengulangan) yang berfokus pada *efektifitas* aplikasi yang dirancang.

G. Desain Sistem

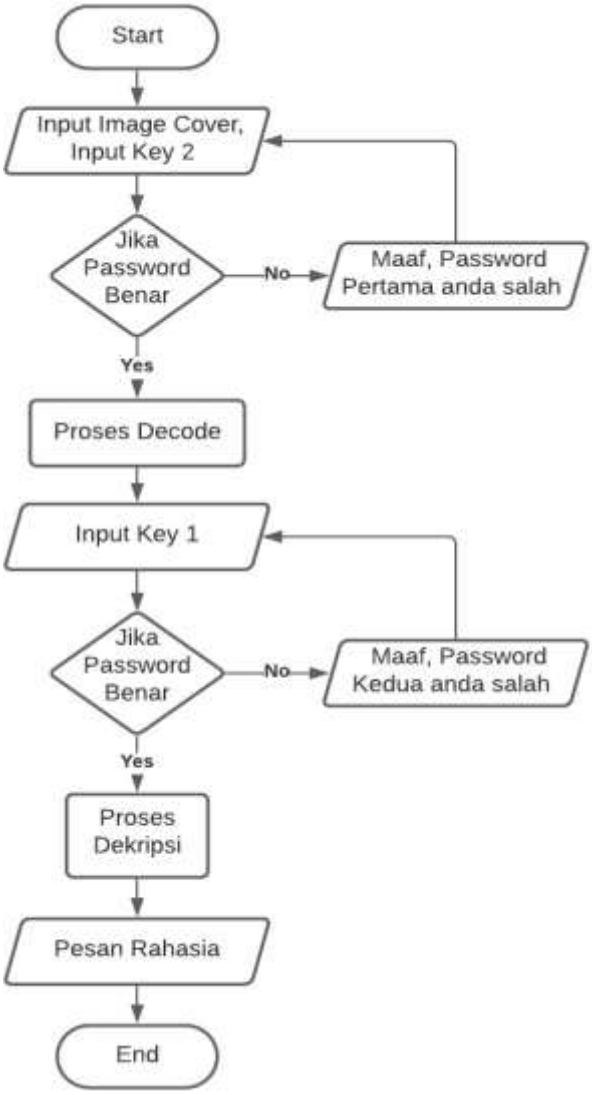
1. Flowchart Sistem



Gambar 3. 1. *Flowchart Menu*

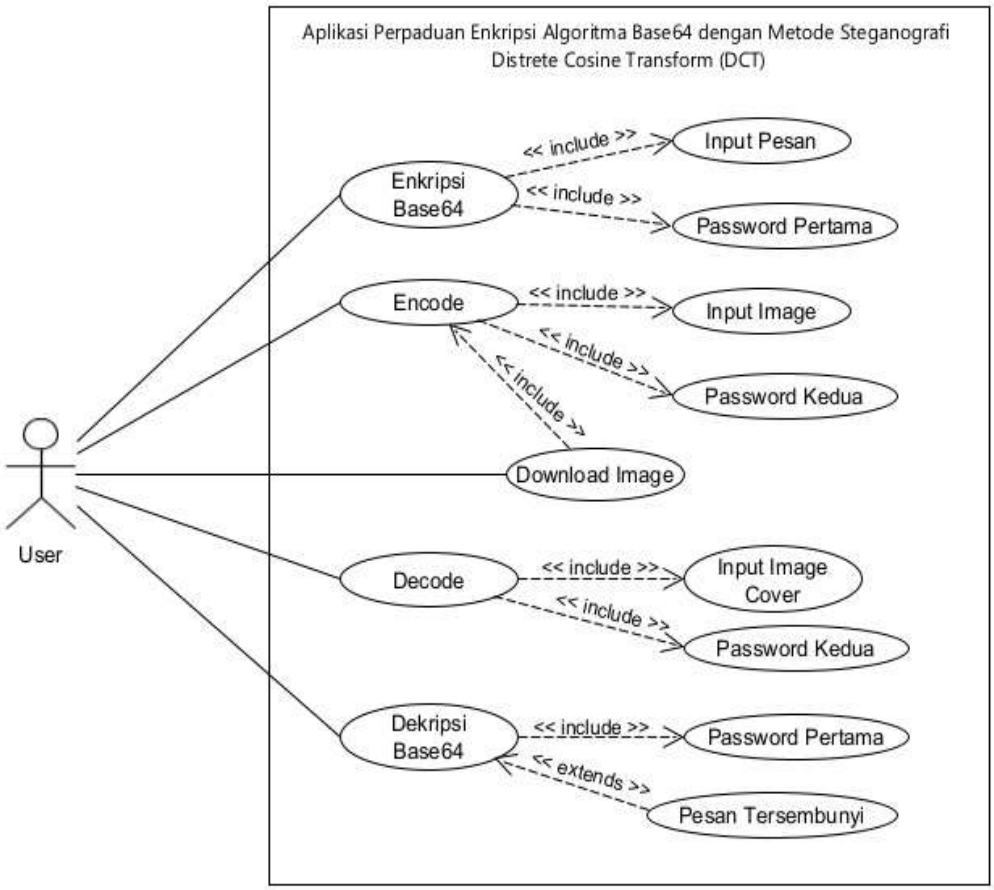


Gambar 3. 2. *Flowchart Encode*



Gambar 3. 3. *Flowchart Decode*

2. *Use Case Diagram*.



Gambar 3. 4. *UseCase Diagram*

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Perancangan Sistem

Pada dasarnya aplikasi yang dibuat dalam tugas akhir ini adalah aplikasi perpaduan *enkripsi* algoritma *base64* dengan metode *stegenografi DCT* yang terdiri dari 2 (dua) unsur utama yakni perangkat keras dan perangkat lunak, sehingga rancangan sistemnya terdiri dari rancangan keras dan rancangan perangkat lunak.

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 2GB.

2. Rancangan perangkat lunak

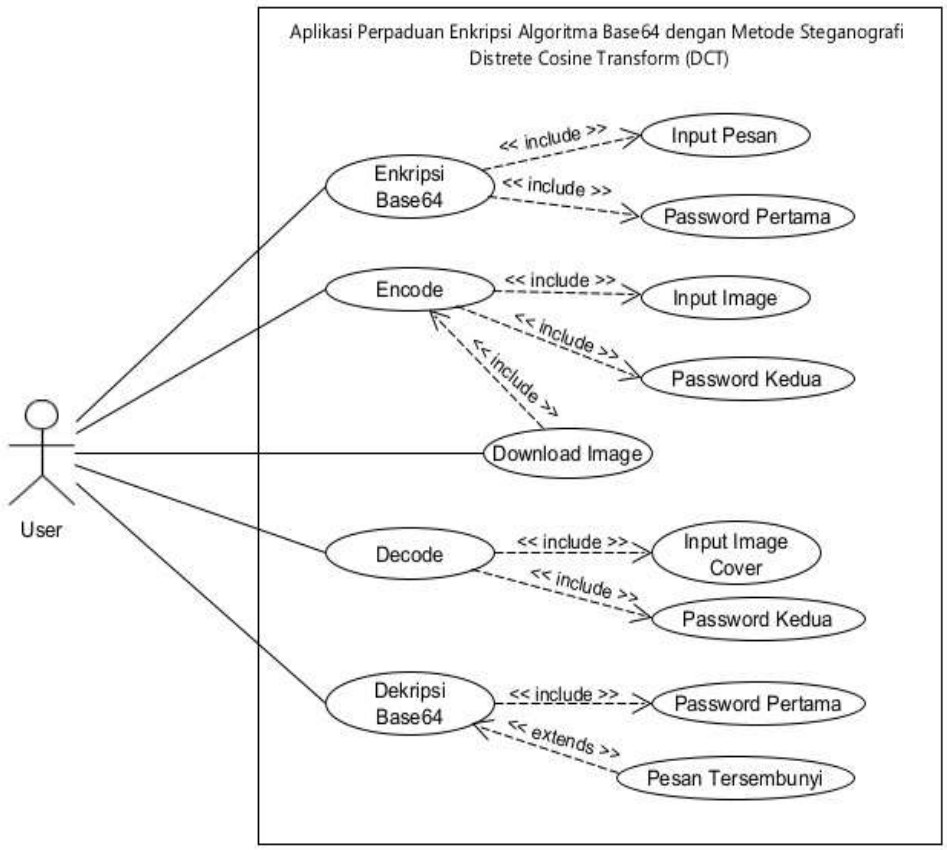
Adapun perangkat lunak yang digunakan untuk mengimplementasikan rancangan pada penelitian ini adalah untuk rancangan bangun perangkat lunak yaitu menggunakan sistem operasi Windows 10 Pro 64 bit, bahasa pemrograman PHP, dan *tools s*eperti *xampp, SublimeText* , *browser* serta jenis *file* yang digunakan saat penelitian adalah *txt* dan *igame*. 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 ke dalam media cover yaitu gambar. Adalapun dalam melakukan perancangan sistem ialah dengan memanfaatkan Diagram UML berupa *Usecase diagram, Activity diagram, dan Seqeuence diagram.*

1) *Usecase diagram*

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



Gambar:4.1.*Usecase Diagram*

Penjelakasan *usecase diagram* :

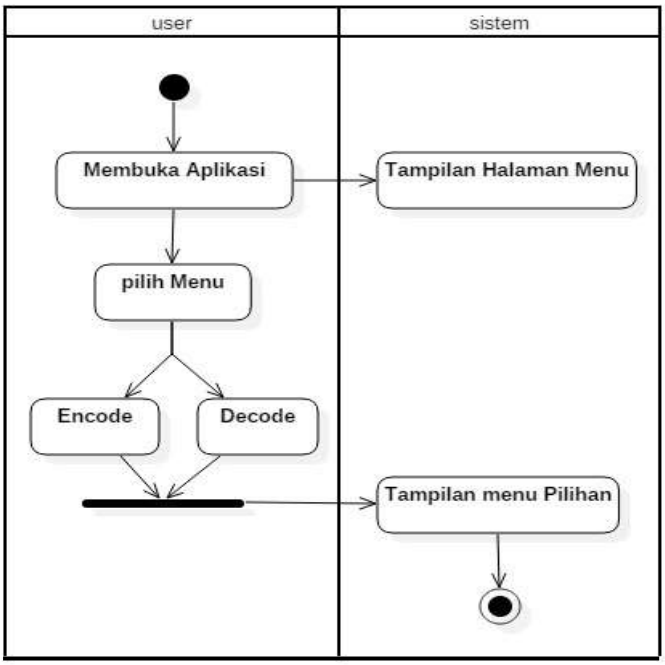
Tabel 4. 1. Penjelasan *usecase* diatas

Nama use case	Deskripsi *use case*
Input pesan *(plaintext)*	*Use case* ini menjelaskan user bisa menginput pesan terlebih dahulu .
Password pertama	*Use case* ini menjelaskan user mengimput Kunci untuk proses *base64*
Input image	*Use case* ini menjelaskan user memilih gambar, gambar sebagai tempat penyisipan tekt.
Passwort kedua	*Use case* ini menjelaskan tentang user melakukan input kunci private sebagai kunci untuk proses *encode* gambar.
Encode/enkripsi	*Use case* ini menjelaskan tentang proses *enkripsi/encode* file format (.txt) menggunakan kunci privat dan algoritma *Base64* dan langsug *stego* tekt ke gambar dengan *DCT*.
Dowload	*use case* ini menjelaskan user mendowload gambat yang telah tersisipkan teks.
Input cover image	*Use case* ini menjelaskan tentang user memilih gambar yang sudah terdowload.
Decode/dekripsi	*Use case* ini menjelaskan proses *decode/dekripsi* teks ke base64 *decode* dengan *dct.*

2) Activity diagram

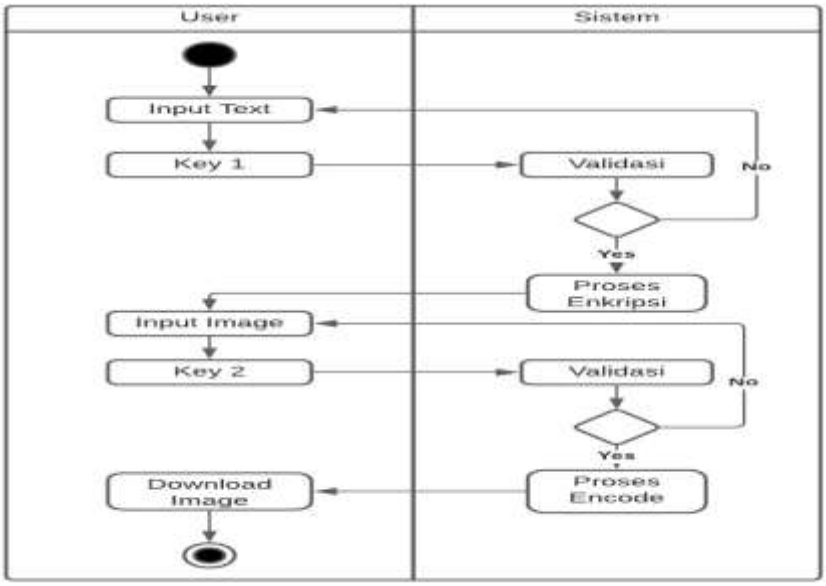
Activity diagram memodelkan alur kerja *(workflow)* sebuah proses dan urutan dalam suatu proses. Activity diagram pada sistem yang penulis rancang antara lain :

1. Rancangan Menu



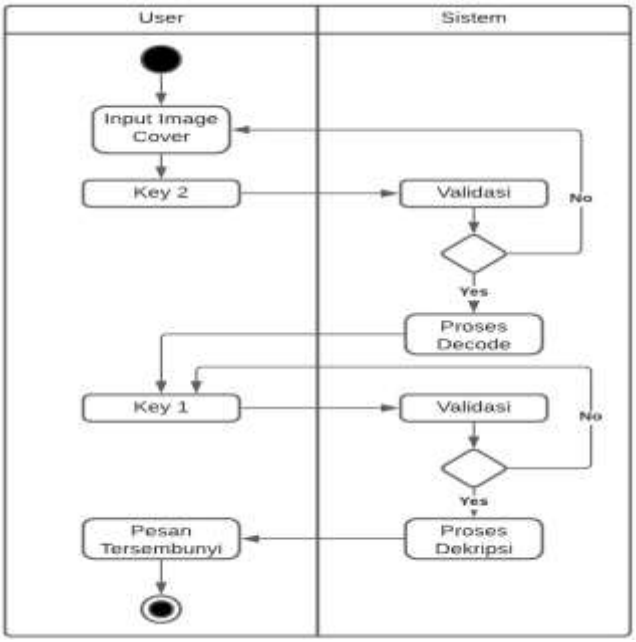
Gambar 4. 2. *Activity diagram* rancangan menu

2. Rancangan Encode



Gambar 4. 3. *Activity diagram* rancangan *encode*

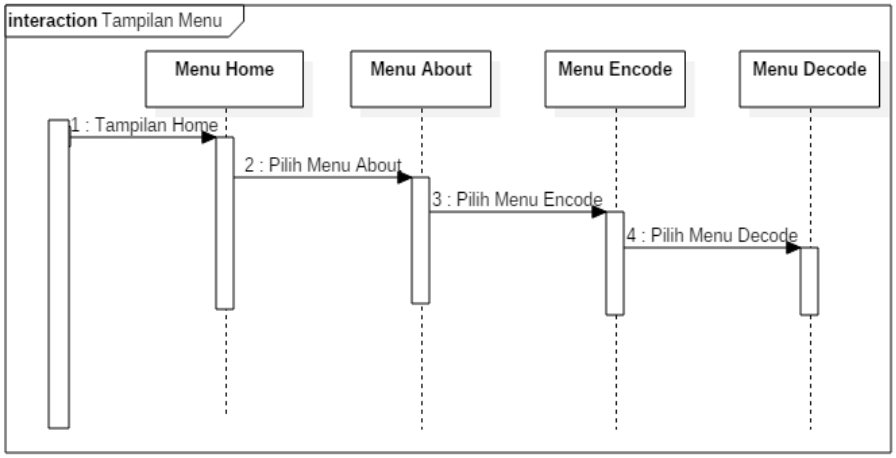
3. Rancangan Decode



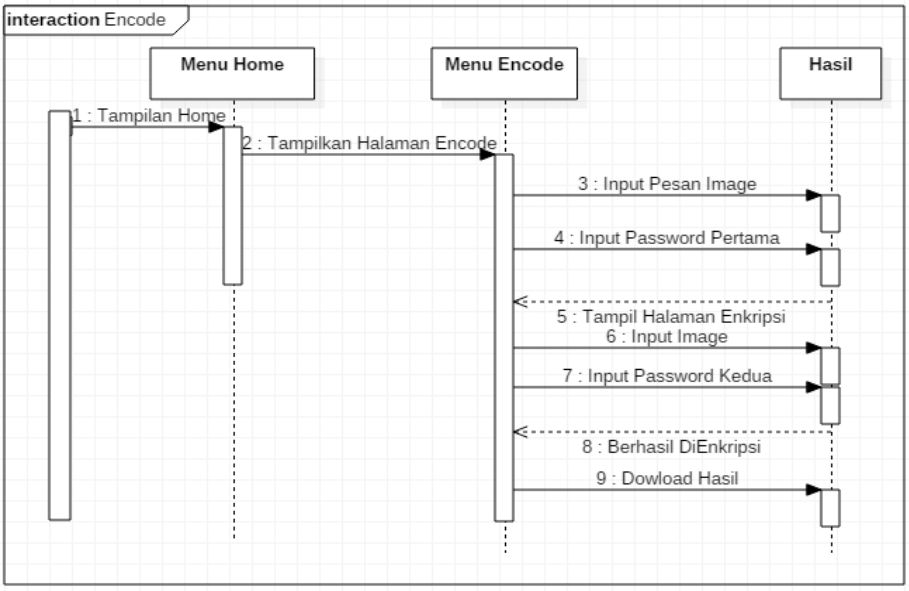
Gambar 4. 4. *Activity diagram* rancangan *decode*

3) *Sequence diagram*

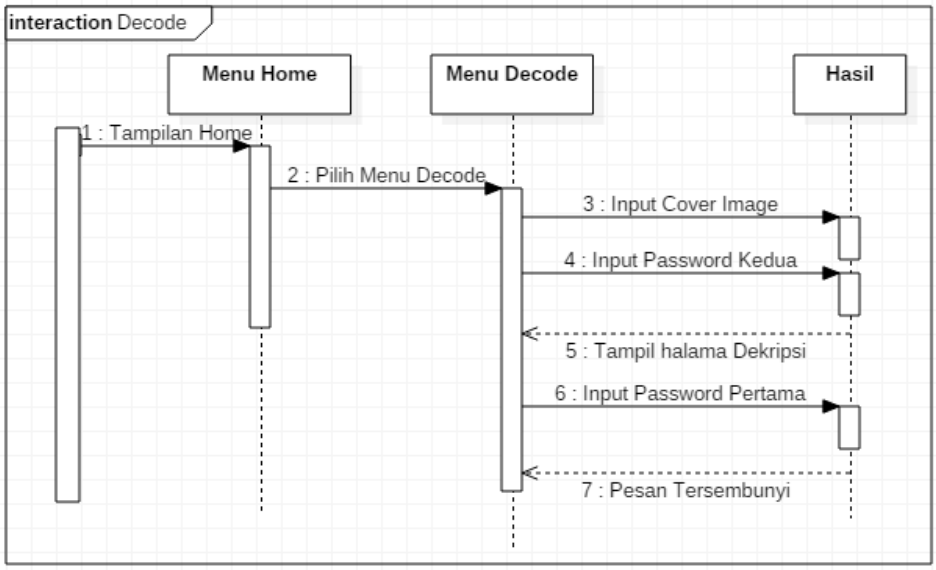
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antara objek dan mengkondisikan komunikasi antara objek-objek tersebut. Berikut squence diagram aplikasi yang akan dibangun.



Gambar 4. 5. *Sequence diagram* Tampilan Menu



Gambar 4. 6. *Sequence diagram* rancangan *Encode*



Gambar 4. 7. *Sequence diagram* rancangan *decode*

b. Perancangan *interface*

Rancangan interface merupakan desain tampilan dari sebuah sistem yang bertujuan untuk menggambarkan sistem yang akan dibangun. Adapun rancangan interface untuk Aplikasi Steganografi Text ke Image dan Enkripsi Base64 dengan metode DCT, yaitu :

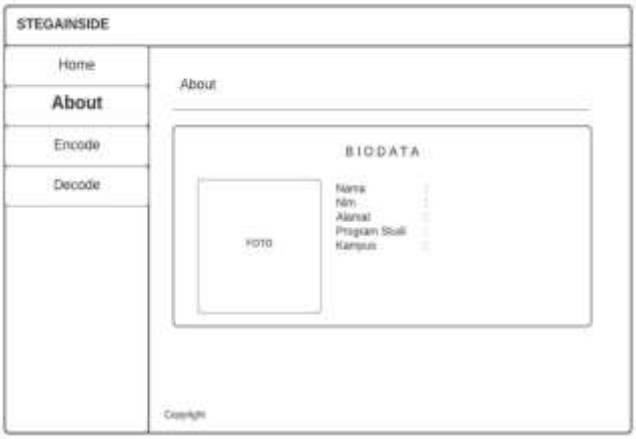
1) Rancangan *interface* Halaman utama



Gambar 4. 8. Rancangan *Interface* Halaman utama

Halaman utama merupakan tampilan awal pada saat aplikasi ini di akses pertama kali, dengan menampilkan nama aplikasi dan kalimat dasar latar belang kenapa mengambil judul ini, *encode* dan *decode*.

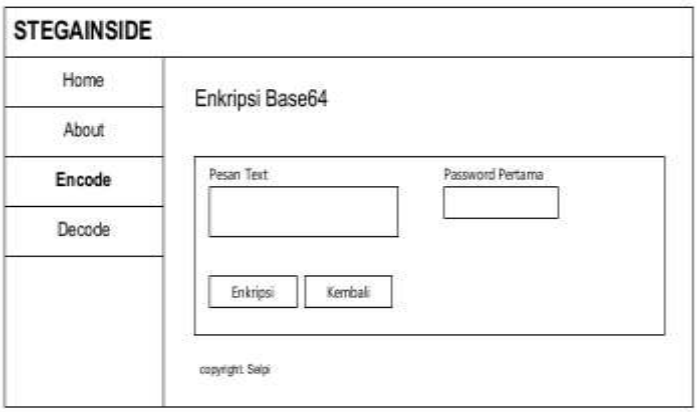
2) Rancangan *interface Menu About*



Gambar 4. 9 Rancangan *Interface* Halaman *Abaut*

Halaman about ini menampilkan tampikab biodata penulis dan juga terdapat foto penulis.

3) Rancangan *interface Menu encode*



Gambar 4. 10. Rancangan *Interface* Halaman *encode*

Menu *Encode* merupakan tampilan yang berfungsi untuk melakukan proses *encode*. Di dalam ini, terdapat fitur yaitu :

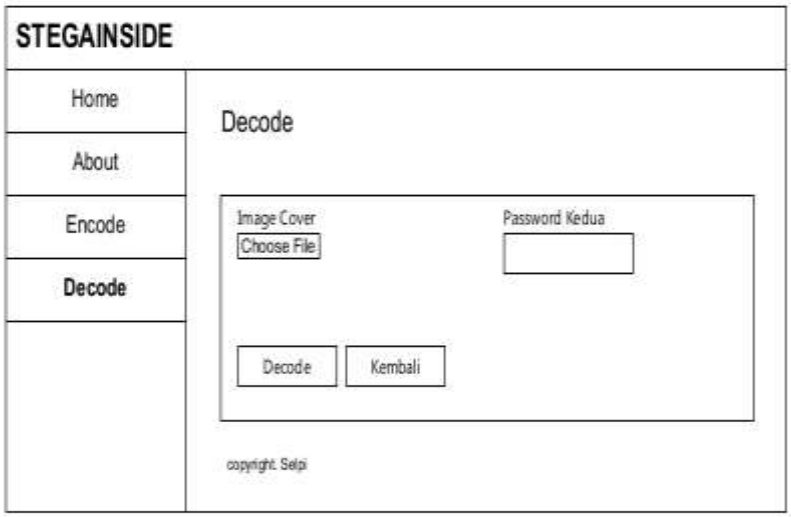
a. Penas teks : tempat untuk memasukkan pesan yang akan di *enkripsi*.

b. Kunci / *password* pertama : tempat untuk memasukkan kunci

c. *Image Cover* : tempat untuk memasukkan *igamge* sebagai *cover* nantinya.

d. *Enkripsi :* tempat melakukan proses *enkripsi*

4) Rancangan *Interface Menu Decode*



Gambar 4. 11. Rancangan *Interface* Halaman *decode*

Menu *Encode* merupakan tampilan yang berfungsi untuk melakukan proses *encode*. Di dalam ini, terdapat fitur yaitu :

a. *Input cover :* pada tampilan ini memasukkan gambar yang sudah di *encode.*

b. Kunci/*passwors* kedua *:* masukkan kunci yang sama pasa *encode.*

c. *Decode :* untuk melakukan proses *decode*

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 sebagai berikut:

1. Tampilan Halaman Utama

Pada tampilan ini merupakan tampilan awal pada saat kita membuka aplikasi, disini terdapat nama aplikasi yaitu *Stegainside*, judul aplikasi, terdapat latar belakan penulis mangambil judul ini, *menu* pilihan *encode* dan *decode*.



Gambar 4.12 Tampilan Halaman Utama

2. Tampilan *Menu About*

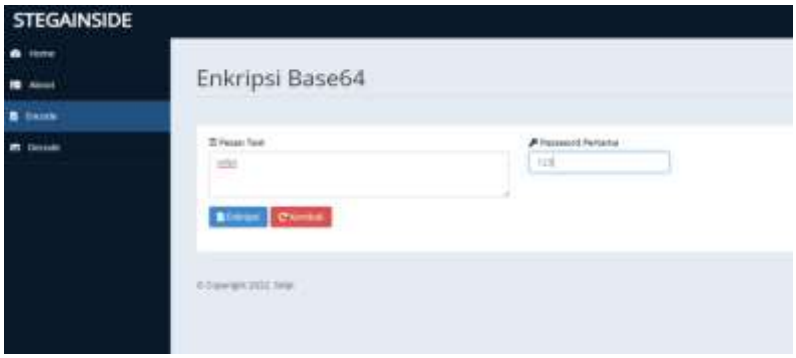
Pada halaman ini menampilkan *menu about* didalamnya terdapat boidata penulis dan juga foto penulis.



Gambar 4.13. Tampilan Menu *About*

3. Tampilan *Menu Encode*

Pada halaman ini menampilkan *menu encode* disini user dapat menginput pesan yang akan di enkripsi menggunakan algoritma *base64*, user juga dapat memasukkan *cover image* dan juga dapat memasukkan kunci, terdapa juga pilihan *encode* jika ingin melanjutkan proses *enkripsi* dan juga pilihan kembali jika tidak ingin melanjutkan.



Gambar 4.14. Tampilan *Menu Encode*



Gambar 4.15. Tampilan proses penyisipan gambar



Gambar 4.16. Tampilan *Download*

4. Tampilan *Menu Decode*

Pada halaman ini user mamasukkan *cover image* yang sudah berisi pesan atau sudah ter*encode* dan juga mamasukkan kunci yang sama pada saat proses *encode.*



Gambar 4.17. Tampilan Proses *Decode*



Gambar 4.18. Tampilan Hasil *Decode*

C. Pengujian Algoritma

Pada tahapan pengujian algoritma penulis melakukan pengujian terhadap algoritma yang telah gunakan. Pengujian yang dilakukan untuk mengetahui apakah algoritma yang digunakan hasilnya sama dengan aplikasi yang telah dirancang. Pada pengujian algoritma ini menggunakan algoritma base64 dan metode steganografi dct.

1. ***Base64***

Tabel 4.2 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	□

Table 4. 3 *Index Base64*

Value	Char		Value	Char		Value	Char		Value	Char
0	A		16	Q		32	g		48	w
1	B		17	R		33	h		49	x
2	C		18	S		34	i		50	y
3	D		19	T		35	j		51	z
4	E		20	U		36	k		52	0
5	F		21	V		37	l		53	1
6	G		22	W		38	m		54	2
7	H		23	X		39	n		55	3
8	I		24	Y		40	o		56	4
9	J		25	Z		41	p		57	5
10	K		26	a		42	q		58	6
11	L		27	b		43	r		59	7
12	M		28	c		44	s		60	8
13	N		29	d		45	t		61	9
14	O		30	e		46	u		62	+
15	P		31	f		47	v		63	/
Pad	=									

Bentuk pengujian algoritma *Base64* penggunaan *plaintext* dapat dilihat pada contoh berikut:

a. Tahapan *Enkripsi*

Langkah proses encoding *Base64* yaitu:

1) Input

Huruf = pio

ASCII = 112 105 111 (mencari kode ASCII dari masing-masing huruf).

Bit = 01110000 01101001 01101111

Deretan binari dijadikan 1 baris panjang

"pio": 011100000110100101101111

(mencari bilangan biner 8 bit dari kode ASCII yang ada kemudian gabungkan 8 bit tersebut).

2) Proses

Bit = 011100 000110 100101 101111

(Hasil dari 8 bit tersebut kemudian diubah kedalam bentuk 6 bit).

Tabel 4. 4 Biner 6 bit

No	Biner 6Bit	Desimal
1	011100	28
2	000110	6
3	100101	37
4	101111	47

Dari tabel diatas didapat nilai desimal 28 2 37 47.

3) Output

Nilai desimal 28 2 37 47 kemudian dirubah kedalam bentuk sesuai dengan tabel yang merupakan index algoritma Base64.

Tabel 4.5 Output Base64

No	Decimal	Index Value
1	28	C
2	6	G
3	37	L
4	47	V

Didapat hasil encoding cGlv, jadi pesan pio di encoding menggunakan Base64 menjadi cGlv.

b. Tahapan Dekripsi

Untuk proses deskripsi atau decoding tidak jauh berbeda dengan proses encoding, hasil encoding kemudian di rubah kedalam bentuk desimal berdasarkan tabel index Base64 dan dilanjutkan dengan mengubah kebentuk biner 6bit yang kemudian digabung menjadi binari 8 bit.

Encoding = cGlv

Desimal = 28 6 37 47

Biner 6 Bit = 011100 000110 100101 101111

Lalu digabung biner 6 Bit nya menjadi 8 Bit

Biner 8 Bit = 01110000 01101001 01101111

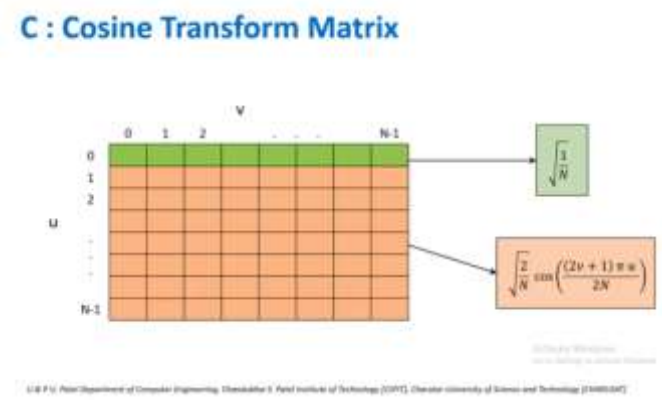
Kemudian hasil binary 8 bit ke kode ASCII , maka hasil teks akan balik ke semula yaitu pio.

2. Distrete Cosine Transform

$$F = C f C^T$$

$$C(u,v) = \begin{cases} \sqrt{\frac{1}{N}} & u=0,\ 0 \le v \le N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2v+1)\pi u}{2N}\right) & 1 \le u \le N-1,\ 0 \le v \le N-1 \end{cases}$$

Gambar 4.18 Rumus



Gambar 4.19 Cosine Transform Matrix

Untuk baris pertama kita mengikuti persamaan pertama yaitu $\sqrt{\frac{1}{N}}$

Karena nilai N = 4 jadi :

$$\sqrt{\frac{1}{N}} = \sqrt{\frac{1}{4}} = 0.5$$

Untuk baris ke dua dan selanjutnya kita menggunakan persamaan ke-2

$$\sqrt{\frac{2}{N}} \cos \left(\frac{(2v+1)\pi u}{2n}\right.$$

Jadi untuk u=1 dan v=0 :

$$\sqrt{\frac{2}{N}} \cos \left(\frac{(2v+1)\pi u}{2n}\right. =$$

$$= \sqrt{\frac{2}{4}} \cos \left(\frac{(2(0)+1)\pi 1}{2.4}\right)$$

$$= \sqrt{\frac{2}{4}} \cos \left(\frac{(1)\pi}{8}\right)$$

$$= \sqrt{\frac{2}{4}} \cos \left(\frac{180}{8}\right)$$

$$= \sqrt{\frac{2}{4}} \cos 22.5$$

$$= 0.71 \, . \, 0.92$$

= 0.65

Cara ini dilakukan untuk pengisian matrix selanjut nya sehingga mengjasilkan :

Tabel 4.6 *Cosine Transform Matrix*

u \ v→	0	1	2	3
0	0.5	0.5	0.5	0.5
1	0.65	0.27	-0.27	-0.65
2	0.5	-0.5	-0.5	0.5
3	0.27	-0.65	0.65	-0.27

Selanjut nya hasil cosine matrix di atas di kalikan dengan spatial domain lalu di kalikan lagi dengan cosine transform.

$F = C \times f \times C^T$

Untuk mencari nilai dari F diatas maka di perlukan langkah-langkah sebagai baeikut:

Tabel 4.7. Matrix C

C=

0.5	0.5	0.5	0.5
0.65	0.27	-0.27	-0.65
0.5	-0.5	-0.5	0.5
0.27	-0.65	0.65	-0.27

x

Tabel 4.8. Matrix F

f =

52	55	61	66
63	59	66	90
62	59	68	113
63	58	71	122

Table 4.9. Matrix C^T

x $\mathbf{C^T}$ =	0.5	0.65	0.5	0.27
	0.5	0.27	-0.5	-0.65
	0.5	-0.27	-0.5	0.65
	0.5	-0.65	0.5	-0.27

Tabel4.10. Matrix hasil F

282	-53.80	33.50	-8.99
-29.24	24.59	-20.26	6.33
-8	1.22	-4.50	1.65
-3	-2.08	-0.79	-1.31

Penyelesaian :

$Cf_{1.1}$ = (0,5 x 52) + (0,5 x 63) + (0,5 x 62) + (0,5 x 63)

= 26 +31,5 + 31 + 31,5 = 120

$Cf_{1.2}$ = (0,5 x 55) + (0,5 x 59) + (0,5 x 59) + (0,5 x 58)

= 27,5 + 29,,5 + 29,5 + 29 = 115,5

$Cf_{1.3}$ = (0,5 x 61) + (0,5 x 66) + (0,5 x 68) + (0,5 x 71)

= 30,5 + 33 + 34 + 35,5 = 133

$Cf_{1.4}$ = (0,5 x 66) + (0,5 x 90) + (0,5 x 113) + (0,5 x 122)

= 33 + 45 + 56,5 + 61 = 195,5

$Cf_{2.1}$ = (0,65 x 52) + (0,27 x 63) + (-0,27 x 62) + (-0,65 x 63)

= 33,8 + 17,01 + (-16,74) + (-40,95) = -6,88

$Cf_{2.2}$ = (0,65 x 55) + (0,27 x 59) + (-0,27 x 59) + (-0,65 x 58)

= 35,75 + 15,93 + (-15,93) + (-37,7) = -1,95

$Cf_{2.3}$ = (0,65 x 61) + (0,27 x 66) + (-0,27 x 68) + (-0,65 x 71)

= 39,65 + 17,82 + (-18,36) + (-46,15) = -7,04

$Cf_{2.4}$ = (0,65 x 66) + (0,27 x 90) + (-0,27 x 113) + (-0,65 x 122)

= 42,9 + 24,3 + (-30,51) + (79,3) = -42,61

$Cf_{3.1}$ = (0,5 x 52) + (-0,5 x 63) + (-0,5 x 62) + (0,5 x 63)

= 26 + (-31,5) + (-31) + 31,5 = -5

$Cf_{3.2}$ = (0,5 x 55) + (-0,5 x 59) + (-0,5 x 59) + (0,5 x 58)

= 27,5 + (-29,5) + (-29,5) + 29 = -2,5

$Cf_{3.3}$ = (0,5 x 61) + (-0,5 x 66) + (-0,5 x 68) + (0,5 x 71)

= 30,5 + (-33) + (-34) + 35,5 = -1

$Cf_{3.4}$ = (0,5 x 66) + (-0,5 x 90) + (-0,5 x 113) + (0,5 x 122)

= 33 + (-45) + (-56,5) + 61 = -7,5

$Cf_{4.1}$ = (0,27 x 52) + (-0,65 x 63) + (0,5 x 62) + (-0,27 x 63)

= 14,04 + (-40,95) + 40,3 + (-17,01) = -3,62

$Cf_{4.2}$ = (0,27 x 55) + (-0,65 x 59) + (0,5 x 59) + (-0,27 x 58)

= 14,85 + (-38,35) + 38,35 + (-15,66) = -0,81

$Cf_{4.3}$ = (0,27 x 61) + (-0,65 x 66) + (0,5 x 68) + (-0,27 x 71)

= 16,47 + (-42,9) + 44,2 + (-19,17) = -1,4

$Cf_{4.4}$ = (0,27 x 66) + (-0,65 x 90) + (0,5 x 113) + (-0,27 x 122)

= 17,82 + (-58,5) + 73,45 + (32-94) = -0,17

Selanjutnya untuk menghasilkan matrix F maka hasil dari perkalian martix C dan f di kalikan dengan matrix C^T maka caranya sebagai berikut:

$F_{1.1}$ = (120 x 0,5) + (115,5 x 0,5) + (133 x 0,5) + (195,5 x 0,5)

= 60 + 57,5 + 66,5 + 97,75 = 282

$F_{1.2}$ = (120 x 0,65) + (115,5 x 0,27) + (133 x -0,27) + (195,5 x -0,65)

= 78 + 31,185 + (-35,91) + (-127,075) = - 53,8

$F_{1.3}$ = (120 x 0,5) + (115,5 x -0,5) + (133 x -0,5) + (195,5 x 0,5)

= 60 + (-57,5) + (-66,5) + 79,75 = 33,5

$F_{1.4}$ = (120 x 0,27) + (115,5 x -0,65) + (133 x 0,65) + (195,5 x -0,27)

= 32,4 + (-0,975) + 86,45 + (-52,875) = -8,99

$F_{2.1}$ = (-6,88 x 0,5) + (-1,95 x 0,5) + (-7,04 x 0,5) + (-42,61 x 0,5)

= (-3,44) + (-9,75) + (-3,52) + (-21,305) = -29,24

$F_{2.2}$ = (-6,88 x 0,65) + (-1,95 x 0,27) + (-7,04 x -0,27) + (-42,61 x 0,65)

= (-4,472) + (-0,5265) + 1,9008 + 27,6965 = 24,59

$F_{2.3}$ = (-6,88 x 0,5) + (-1,95 x -0,5) + (-7,04 x -0,5) + (-42,61 x 0,5)

= (-3,44) + 0,975 + 3,54 + (-21,305) = -20,26

$F_{2.4}$ = (-6,88 x 0,27) + (-1,95 x -0,65) + (-7,04 x 0,5) + (-42,61 x -0,27)

= 1,8576 + 1,2677 + (-4,576) + 11,5047 = 6,33

$F_{3.1}$ = (-5 x 0,5) + (-2,5 x -0,5) + (-1 x 0,5) + (-75x 0,5)

= (-2,5) +(-1,25) + (-0,5) + (-3,75) = -8

$F_{3.2}$ = (-5 x 0,6) + (-2,5 x -0,27) + (-1 x -0,27) + (-75x -0,65)

= (-3,25) + (-0,675) + 0,27 + 4,875 = 1,22

$F_{3.3}$ = (-5 x 0,5) + (-2,5 x -0,5) + (-1 x -0,5) + (-75x 0,5)

= (-2,5) + 1,25 + 0,5 + (-3,75) = -4,5

$F_{3.4}$ = (-5 x 0,27) + (-2,5 x -0,65) + (-1 x 0,65) + (-75x -0,27)

= (-1,35) + 1,625 + (-0,625) + 2,025 = 1,65

$F_{4.1}$ = (-3,62 x 0,5) + (-0,8 x -0,5) + (-1,4 x 0,5) + (-0,17 x 0,5)

= (-1,81) + (-0,4) + (-0,7) + (-0,08) = -2,99

$F_{4.2}$ = (-3,62 x 0,65) + (-0,8 x 0,27) + (-1,4 x -0,27) + (-0,17 x -0,65)

= (-2,353) + (-0,216) + 0,378 + 0,1105 = -2,08

$F_{4.3}$ = (-3,62 x 0,5) + (-0,8 x -0,5) + (-1,4 x -0,5) + (-0,17 x 0,5)

= (-1,81) + 0,4 + 0,7 + (-0,85) = -0,795

$F_{4.4}$ = (-3,62 x 0,27) + (-0,8 x -0,65) + (-1,4 x 0,65) + (-0,17 x -0,27)

= (-0,9774) + 0,5265 + (-0,9) + 0,0459 = 1,315

3. Least Significant Bit (LSB)

Contoh pengujian penyisipan sebuah pesan ke dalam citra *digitaltrue color*

a. Misalnya akan menyisipkan karakter cGlv
b. Karakter tersebut terlebih dahulu dikonversi menjadi kode ASCII yang menghasilkan bilangan desimal yaitu **99 71 108 118**
c. Nilai ASCII (dalam bentuk desimal) akan dikonversi menjadi bilangan biner 8 bit sehingga menghasilkan
    a. 99 = 01100011
    b. 71 = 01000111
    c. 108 = 01101100
    d. 118 = 01111010
d. Kemudian mengambil nilai RGB dari tiap-tiap *pixel* gambar yang akan disisipkan pesan, seperti gambar di bawah ini :

px1	px2	px3	px4	px5	px6	px7	px8
px9	px10	px11	px12	px13	px14	px15	px16
px17	px18	px19	px20	px21	px22	px23	px24

Gambar 4.20. Ilustrasi *pixel* sebuah gambar

Misalkan data-data RGB tiap *pixel* adalah sebagai berikut :

1. px1 → R = 254, G = 37 dan B = 48
2. px2 → R = 254, G = 27 dan B = 38
3. px3 → R = 225, G = 140 dan B = 146
4. px4 → R = 254, G = 95 dan B = 103
5. px5 → R = 254, G = 52 dan B = 63
6. px6 → R = 255, G = 168 dan B = 172
7. px7 → R = 208, G = 188 dan B = 235
8. px8 → R = 183, G = 240 dan B = 206
9. px9 → R = 40, G = 202 dan B = 105
10. px10 → R = 104, G = 52 dan B = 190

Kemudian nilai-nilai RGB diatas akan diubah menjadi nilai biner, sehingga menghasilkan nilai–nilai berikut:

1. px1 → R = 11111110, G = 00100101, B = 00110000
2. px2 → R = 11111110, G = 00011011, B = 00100110
3. px3 → R = 11100001, G = 01101000, B = 10010010
4. px4 → R = 11111110, G = 01011111, B = 01100111
5. px5 → R = 11111110, G = 00110100, B = 00111111
6. px6 →R = 11111111, G = 10101000, B = 10101100
7. px7 → R = 11010000, G = 10111100, B = 11101011
8. px8 → R = 10001111, G = 00010100, B = 10000100
9. px9 → R = 00101000, G = 11001010, B = 01101001

10. px10 → R = 01101000, G = 00110100, B = 10111110

Masing-masing bit akan disisipkan untuk menggantikan bit ke 8 dari setiap *pixel*, karena 1 *pixel* terdiri dari komponen warna RGB, maka dalam 1 *pixel* dapat menampung 3 bit.Sehingga representasi biner dari *file* gambar telah mengalami perubahan seperti gambar di bawah ini :

Tabel 4.11. Ilustrasi Penyisiapn Pesan RGB

Pixel	**R**	**G**	**B**
px1	1111111**0**	0010010**1**	0011000**1**
px2	1111111**0**	0001101**0**	0010011**0**
px3	1110000**1**	0110100**1**	1001001**0**
px4	1111111**1**	0101111**0**	0110011**0**
px5	1111111**0**	0011010**1**	0011111**1**
px6	1111111**1**	1010100**0**	1010110**1**
px7	1101000**1**	1011110**0**	1110101**1**
px8	1000111**1**	0001010**0**	1000010**0**
px9	0010100**0**	1100101**1**	0110100**1**
px10	0110100**1**	0011010**1**	1011111**0**

B. Pengujian Sistem

Pada tahapan pengujian sistem 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 error maka program bisa di perbaiki.

Pengujian aplikasi dilakukan dengan menggunakan metode pengujian yaitu pengujian blackbox.

1. Pengujian *Blackbox*

Pengujian *Blackbox (blackbox testing)* adalah salah satu metode pengujian yang berfungsi pada sisi *fungsionalitas* yang ada dalam sistem. Kemudian membandingkan hasil keluaran sistem dengan hasil yang diharapkan. Bila hasil yang diharapkan sesuai dengan hasil pengujian, artinya aplikasi sesuai dengan desain yang telah ditentukan sebelumnya. Jika belum sesuai maka perlu dilakukan pengecekan lebih lanjut dan perbaikan.

a. *Blackbox Encode* Berhasil

Tabel 4. 12. Jika *Encode* Pesan Berhasil

Test	Hasil	Kesimpulan
Jika pesan berhasil di *encode* pada *menu encode*	✓	Berhasil, karena tidak ada muncul pesan validasi.
Screenshot		



Gambar 4.21. pesan berhasil di *Encode*

b. *Blackbox Format Image* Bukan PNG

Tabel 4.13. *Format Image* Bukan PNG

Test	Hasil	Kesimpulan
Jika *Format Image* Bukan JPG	✓	Berhasil, karena ada muncul pesan validasi.
Screenshot		
		

c. *Blackbox Password*/Kunci tidak di Isi pada Menu *Encode*

Tabel 4. 14. *Password*/Kunci tidak di Isi

Test	Hasil	Kesimpulan
Password/Kunci tidak di Isi	✓	Berhasil, karena ada muncul pesan validasi.
Screenshot		
		

Gambar 4.23. *Password*/Kunci tidak di Isi

d. *Blackbox Password*/Kunci Salah pada *Menu Decode*

Tabel 4. 15. *Password*/Kunci Salah pada *Menu Decode*

Test	Hasil	Kesimpulan
Password/Kunci Salah pada *Menu Decode*	✓	Berhasil, karena ada muncul pesan validasi.
Screenshot		
		

Gambar 4.24. *Password*/Kunci Salah pada *Menu Decode*

Tabel 4.16 Pengujian Encode

Image Sebelum Encode	**Image Setelah Encode**	**Ukuran image Awal -Akhir**	**Berhasil / Gagal**
 Matahari.png	 Matahari.png	101- 89.7 KB	√
 Pelangi.png	 Pelangi.png	92.1 - 79.7 KB	√
 Mawar.png	 Mawar.png	63.8 - 59.7 KB	√
 Angrek.png	 Angrek.png	102 - 88.9 KB	√
 Mawar.png	 Mawar.png	89.5 - 75.1 KB	√

Tabel 4.17 Pengujian Decode

Image Sebelum Decode	**Image Setelah Decode**	**Pesan Tersembunyi**	**Berhasil / Gagal**
 Daisy.png	 Daysi.png	Pergi	√
 Teratai.png	 Teratai.png	Aku	√
 Matahari.png	 Mataharir.png	Belajar	√
 Angrek.png	 Angrek.png	Membaca	√
 Kenop.png	 Kenop.png	Menulis	√

Tabel 4.18. Selisih ukuran encode

Image Sebelum Encode	Image Setelah Encode	Ukuran Image (kb)	Selisih Penurunan (kb)
 Matahari.png	 Matahari.png	101- 89.7 KB	11.3 KB
 Pelangi.png	 Pelangi.png	92.1 - 79.7 KB	12.4 KB
][=  Mawar.png	 Mawar.png	63.8 - 59.7 KB	4.1 KB
 Angrek.png	 Angrek.png	102 - 88.9 KB	13.1 KB
 Mawar.png	 Mawar.png	89.5 - 75.1 KB	14.4 KB

Image Sebelum Encode	Image Setelah Encode	Ukuran gambar (kb)	Selisih Ukuran(kb)
 Teratai.png	 Teratai.png	69.3 - 65.2 KB	4.1KB
 Angrek.png	 Angrek.png	88.7 - 79.0 KB	9.7KB
 T.Merah.png	 T.Merah.png	100 - 93.9 KB	6.1KB
 Sakura.png	 Sakura.png	103 - 94.7 KB	8.3 KB
 Edible.png	 Edible.png	79.5 - 65.6 KB	13.9 KB
Rata rata			**9.71 KB**

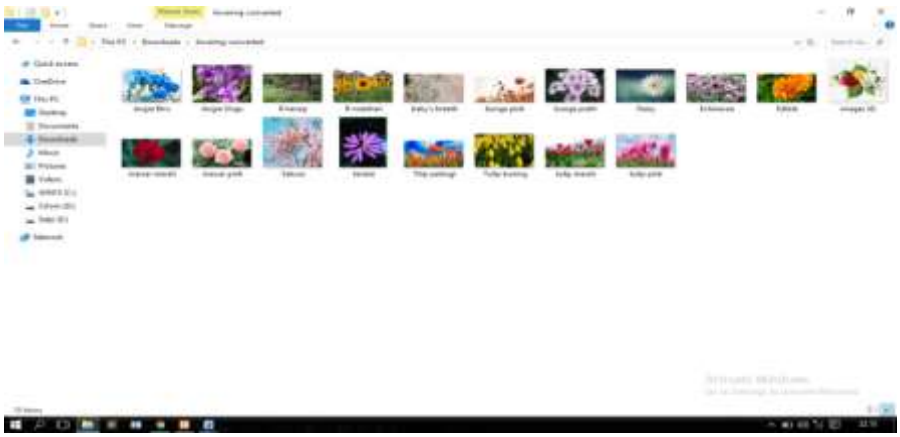
Dari pengujian tabel di atas dapat di lihat bahwa jumlah rata-rata selisih hasil kompresi sebelum dan setelah encode yaitu 9.71 kb.

Table 4.19 Pengujian jpg

Nama Gambar	Ukuran Gambar	Ukuran Pixsel Gambar	Berhasil/Gagal
 Matahari.jpg	9.20 KB	204 x 247 px	-
 Daisy.jpg	41.9 KB	600 x 400 px	-
 Kamboja.jpg	28.3 KB	607 x 438 px	-
 Lavender.jpg	11.9 KB	273 x 185 px	-

Pada pengujian tabel diatas gambar yang digunakan adalah gambar berformat jpg,pada format ini secara keseluruhan gaga karena gambar *cover* yang digunakan pada teknik steganografi ini adalah file gambar berekstensi png.

Pada gambar dibawa menampilkan hasil gambar pada folder berhasil tampil dengan baik baik.



Gambar 4.25 Pengujian Hasil

Tabel 4.20. Hasil Pengujian

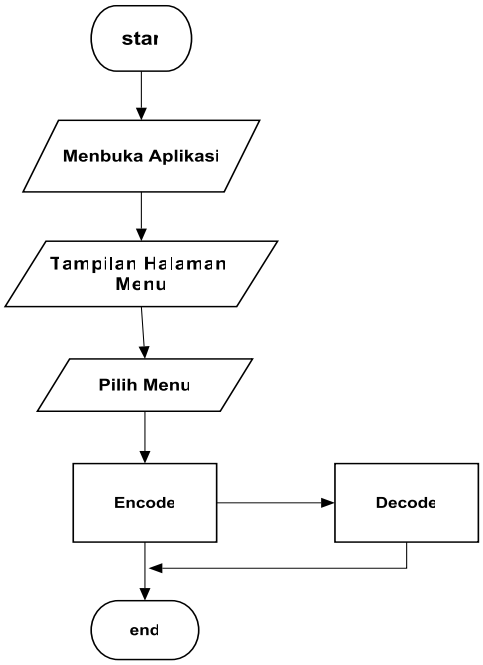
No	Modular	Berhasil	Gaga
1	Tampilan awal aplikasi	√	-
2	Tampilan input pesan/plaintext	√	-
3	Tampilan input gambar cover	√	-
4	Tampilan input kunci/ key	√	-
5	Tampilan validasi	√	-
6	Tampilan akhir gambar (folder)	√	-
7	Proses tombol encode	√	-
8	Proses tombol kembali	√	-
9	Proses decode	√	-
10	Proses download gambar	√	-
11	Tampilan hasil decode	√	-
Total		11	0

Dari tabel diatas dapat disimpulkan pengujian berhasil dilakukan dan menghasilkan hasil yang bebas dari kesalahan. Aplikasi berjalan sesuai yang dirancang dan sesuai dengan yang diharapkan, baik dalam tampilan maupun penulisan code pemrograman.

2. Pengujian White*box*

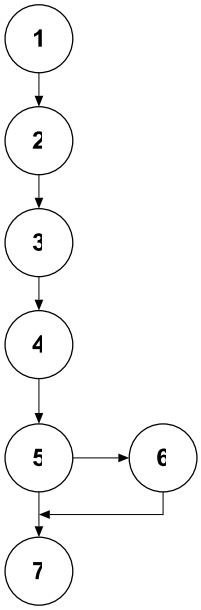
Pada tahap pengujian *white box*, dilakukan pengujian terhadap *method source code* yang digunakan pada aplikasi ini. Pengujian *white box* dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan lancar dan tidak terdapat kesalahan dalam logika pemrograman.

a. Flowchart dan Flowgraph



Gambar 4. 26. *Flowchart* Tampilan Menu

Dari *flowchart Iogin* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4. 27. *Flowgraph* Tampilan *Menu*

Dari *flowgraph* menu *login* pada atas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexcity* V(G) asal *Egde* dan *Node*:

   Menggunakan rumus : V(G) = E – N + 2

   E (*edge*) = 7

   N (*Node*) = 7

   P (Predikat *Node*) = 1

   Penyelesaian : V(G) = E – N + 2

   = 7 – 7 + 2

   = 2

Predikat (P) = P + 1

= 1 + 1

= 2

2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas mempunyai Region = 2

3) *Independentpath* pada *flowgraph* di atas ialah:

Path 1 = 1 – 2 – 3 – 4 – 5– 6 – 7

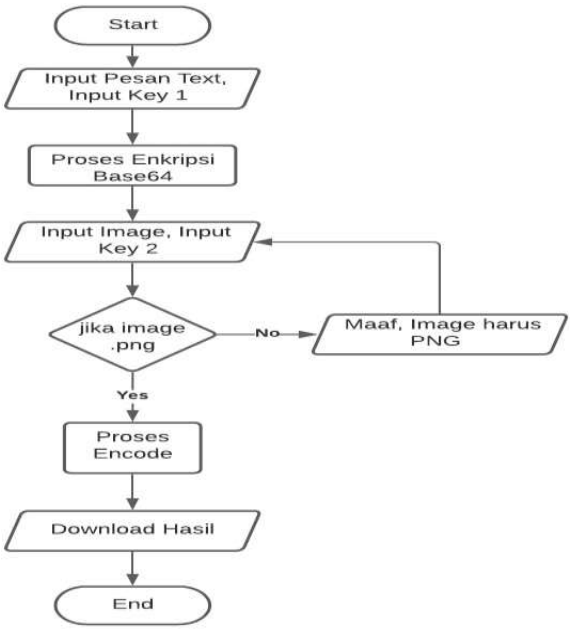
Path 2 = 1 – 2 – 3 – 4 – 5 – 7

4) Grafik Matriks Aktivitas *Menu*

Tabel 4.21. Grafik Matriks Aktivitas *Menu*

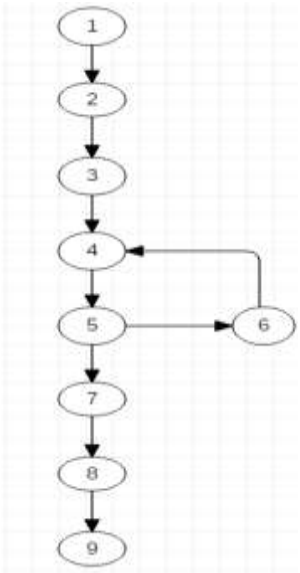
	1	2	3	4	5	6	7	E – 1
1		1						1 – 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0
4					1			1 – 1 = 0
5						1	1	2 – 1 = 1
6						1		1 – 1 = 0
7								0
SUM(E+1)								1 + 1 = 2

b. *Flowchart dan Flowgraph Encode*



Gambar 4.28 . *Flowchart Encode*

Dari *flowchart encode* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4. 29. *Flowgraph Encode*

Dari *flowgraph encode* pada atas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexcity* V(G) asal *Egde* dan *Node*:

   Menggunakan rumus : V(G) = E – N + 2

   E (*edge*) = 9

   N (*Node*) = 9

   P (Predikat *Node*) = 1

   Penyelesaian : V(G) = E – N + 2

   = 9 – 9 + 2

   = 2

   Predikat (P) = P + 1

   = 1 + 1

   = 2

2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas mempunyai Region = 2

3) *Independentpath* pada *flowgraph* di atas ialah:

   Path 1 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9

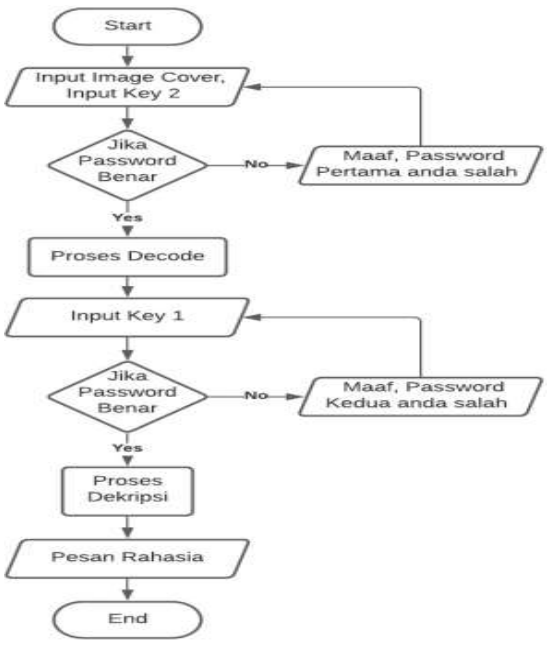
   Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 4 – 5 – 6 – 7 – 8 – 9

4) Grafik Matriks Aktivitas *Encode*

Tabel 4.22. Grafik Matriks Aktivitas *Encode*

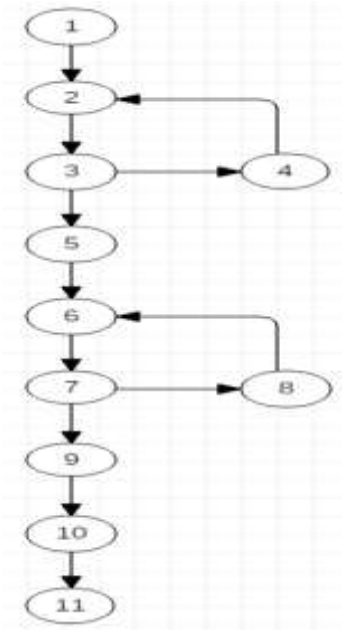
	1	2	3	4	5	6	7	8	9	E – 1
1		1								1 – 1 = 0
2			1							1 – 1 = 0
3				1						1 – 1 = 0
4					1					1 – 1 = 0
5						1	1			2 – 1 = 1
6				1						1 – 1 = 0
7								1		1 – 1 = 0
8									1	1 – 1 = 0
9										0
SUM(E+1)										1+ 1 = 2

c. *Flowchart dan Flowgraph Decode*



Gambar 4.30 . *Flowchart Decode*

Dari *flowchart decode* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4. 31. *Flowgraph Decode*

Dari *flowgraph decode* pada atas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexcity* V(G) asal *Egde* dan *Node*:

Menggunakan rumus : V(G) = E – N + 2

E (*edge*) = 12

N (*Node*) = 11

P (Predikat *Node*) = 2

Penyelesaian : V(G) = E – N + 2

= 12 – 11 + 2

= 3

Predikat (P) = P + 1

= 2 + 1

= 3

2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas mempunyai Region = 2

3) *Independentpath* pada *flowgraph* di atas ialah:

Path 1 = 1 – 2 – 3 – 5– 6 – 7 – 9 – 10 – 11

Path 2 = 1 – 2 – 3 – 4 – 2 – 5 – 6 – 7 – 9 – 10 – 11

Path 3 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 6 – 7 – 8 – 9 – 10 – 11

4) Grafik Matriks Aktivitas *Decode*

Tabel 4.23. Grafik Matriks Aktivitas *Decode*

	1	2	3	4	5	6	7	8	9	10	11	E – 1
1		1										1 – 1 = 0
2			1									1 – 1 = 0
3				1	1							2 – 1 = 1
4		1			1							2 – 1 = 1
5						1						1 – 1 = 0
6							1					1 – 1 = 0
7								1				1 – 1 = 0
8						1						1 – 1 = 0
9										1		1 – 1 = 0
10											1	1 – 1 = 0
11												0
SUM(E+1)												1 + 2 = 3

d. Hasil Pengujian

Tabel 4.24. Hasil Pengujian

No	*Flowchart*	*Independent Path*	*Region*	*Cyclomatic Complexity*
1	*Menu*	2	2	2
2	Aktivitas *Encode*	2	2	2
3	Aktivitas Decode	3	2	2

BAB V

PENUTUP

A. Kesimpulan

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

Pembuatan aplikasi perpaduan *enkripsi* algoritma *base64* dengan metode *steganografi distrete cosine transporm (DCT)* dalam menyisipkan pesan ke dalam gambar yang berformat *png*. Sistem ini dibangun menggunakan aplikasi *Sublame Text* 3 sebagai perancang aplikasinya dan tampilannya menggunakan *Web Browser*.

File gambar yang telah di *encode* setelah di download akan tersimpan di penyimpanan yang tersedia adapun pegujian sistemnya dilakukan pengujian berulang kali untuk mendapatkan hasil sesuai dengan yang telah ditargetkan.

Aplikasi ini dibuat dengan tujuan untuk dapat membantu pengguna/*user* untuk mengirim sebuah pesan teks yang bersifat rahasia, agar pesan dapat sampai ke tangan penerima tanpa menimbulkan kecurigaan pada pihak lain.

Hasil penelitian menunjukan bahwa *encode* dan *decode* dilakukan dengan pengujian tabel sebanyak 10 kali, dimana terdapat perubahan yang terletak pada berkurang nya ukuran size sebelum dan setelah *encode*, adapun rata-rata penurunan size yaitu 9.71 KB. Dan untuk resolusi sebelum dan sesudah *encode* juga mengalami perubahan.

B. Saran

Adapun saran yang dapat diberikan untuk pengembangan pada penelitianini khususnya untuk keamanan informasi pada pesan yang tentunya sangat diperlukan yaitu :

1. Pengembangan bisa melakukan pengujian tidak hanya pada gambar yang format.png , tetapi bisa menambahkan format lainnya seperti *jpg,bpm* sebagai *cover image.*
2. Pengembangan dapat menambahkan tipe data berformat*doc, pdf,* dan *rar*.
3. Untuk segi penyisipan, diharapkan pengembangan mampu menyisipkan data/pesan ke dalam media audio ataupun media video.
4. Diharapkan ada pengembangan selanjutnya pada algoritma yang digunakan, tidak hanya *Base64* dan *dct* namun dikembangkan juga dengan menggunakan algoritma lainnya.

DAFTAR PUSTAKA

Ariyus, D. 2008. Pengantar ilmu kriptografi: teori analisis & implementasi. Penerbit Andi.

Adiria. 2010. Analisis dan Perancangan Aplikasi Steganografi pada Citra Digital menggunakan Metode LSB(*Least Significant Bit*). Skripsi. Jakarta: Universitas Islam Negeri Syarif Hidayatullah.

Darma B. 2019. Implementasi Algoritma Base64 untuk Mengamakan SMS pada Smartphone. *Jurnal Informatika.*

Darmayanti,A. H. 2016. Sistem Steganografi Pada Citra Digital Menggunakan *Least Significant Bit*.In *Prosiding Seminar Sains*dan *Teknologi FMIPA Unmul, Vol. 1, No. 1.*

Didik. 2008. Pengantar ilmu kriptografi: teori analisis, sistem & implementasi. Penerbit Andi.

Donald E. K 1973. Pengertian Algoritma .penerbit Andi.

Haviluddin. 2011. Memahami Penggunaan UML (Unified Modelling Language). *Jurnal Informatika Mulawarman*, 1-15.

Herianto, D. 2013. Penertian Base64,Tabel Ascii

Jubliee. 2009. Penjelasan *Google Chrome*

Minarni, R. 2019. Implementasi Algoritma Base64 Untuk Mengamakan Sms Pada Smartphone. *Building of Informatics, Technology and Science (BITS), 1*(1), 28-33.

Munir, R. 2006. Diktat Kuliah pengerian algoritma.

Prasetio. 2010. Penjelasan dan dan pengertian *HTML (Hypertext Markup Language)*

Rahman, A. (2014). Simbol simbol pada Flowchart dan penjelasannya.

Ridwan M & Yustikasari Y. 2020. Implementasi Steganografi Text to Image JPEG Menggunakan Discrete Cosine Transfrom. *Jurnal Informatika.*

Sadikin. 2012. Pengertian stegaografi : teori steganografi dan inplementasi. Penerbit Andi.

Sitorus, L. 2015. *Algoritma dan Pemrograman*. Penerbit Andi.

Supono & Putratama. 2016 . Pengertian dan Penjelasan PHP (Hypertext Preprocessor)

Yustikasari. 2020. Implementasi Steganografi Text to Image JPEG Menggunakan Discrete Cosine Transfrom. *Jurnal Informatika.*

Wicaksono, Y. 2008. Penjelasan XMPP dan Mysql. *Jurnal Informatika.*

Wati, L. (2013). Penjelasan dan fungsi Xampp. *Yogyakarta: Fasilkom Mercu Buana*.