"PERANCANGAN SISTEM APLIKASI KEAMANAN INFORMASI MENGGUNAKAN *IMAGE* DENGAN METODE *LSB*"

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir Memperoleh

Gelar Sarjana Komputer Pada Fakultas Sains Dan Teknologi

Universitas Pembangunan Panca Budi

Medan

SKRIPSI

OLEH

NAMA : VIJAY SEMBIRING

NPM : 1514370609

PROGRAM STUDI : SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS PEMBANGUNAN PANCA BUDI

MEDAN

2019

ABSTRAK

VIJAY SEMBIRING

PERANCANGAN SISTEM APLIKASI KEAMANAN INFORMASI MENGGUNAKAN IMAGE DENGAN METODE LSB

Perkembangan teknologi informasi saat ini, semakin memudahkan para pelaku kejahatan komputer, dengan menyalah gunakan teknologi tersebut untuk mendukung kegiatannya, dimana aktivitas mereka sangat mengganggu privasi seseorang.di skripsi ini penyisipan pesan *teks* dengan motode *least Significant Bit.* Oleh karena itu diperlukan sebuah sistem atau aplikasi yang aman sehingga dapat mempersulit para pelaku kejahatan komputer untuk melakukan aktivitasnya, dan membantu para pengguna teknologi dalam hal pengamanan data yang diakses tersebut.Untuk mempersulit para pelaku kejahatan komputer maka penulis menggabungkan *kriptografi Caesar Cipher* dengan metode *Least Significant Bit*, yang diharapkan mampu menambah keamanan sebuah pesan teks rahasia.. Untuk meningkatkan keamanan data yang akan disimpan, data yang disimpan juga dienkripsi terlebih dahulu.

Kata Kunci : *Least Significant Bit, Enkripsi, Dekripsi*

DAFTAR ISI

Halaman

DAFTAR GAMBAR

No	Judul	Hal

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Steganografi adalah seni dan ilmu untuk menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui. Berbeda dengan kriptografi yang merahasiakan makna pesan namun keberadaan pesan tetap ada, steganografi merahasiakan dengan menutupi atau menyembunyikan pesan (Harjo, 2016).Steganografi pada dunia digitalisasi telah banyak diterapkan untuk mengirimkan pesan atau informasi rahasia.Tidak hanya itu steganografi juga sering digunakan untuk kegiatan pengarsipan dimana data-data digital di sembunyikan kedalam berkas-berkas digital lainnya yang lebih umum sehingga tidak menarik perhatian pihak-pihak yang ingin mencuri informasi.

Steganografi seiring dengan perkembangannya telah melahirkan berbagai teknik dan metode yang berbeda – beda. Pada berkas digital multimedia seperti citra digital, metode steganografi yang paling umum digunakan adalah metode *Least Significant Bit* atau *LSB* (Singh & Singh, 2015). Metode *LSB* (*Least Significant Bit*) merupakan salah satu metode steganografi dalam teknik domain spasial.Metode *LSB* merubah nilai komponen warna *bit* terakhir dengan *bit* pesan yang akan disembunyikan sehingga menghasilkan citra yang mirip dengan aslinya. Metode ini dapat dikembangkan pada penyembunyian pesan rahasia.

Berdasarkan penjabaran diatas maka penulis bermaksud untuk melakukan penelitian dengan mengembangkan aplikasi pengamanan penyembunyian pesan teks pada citra digital menggunakan metode *LSB* dengan judul "**Perancangan Sistem Aplikasi Keamanan Informasi Menggunakan Image Dengan Metode LSB**".

1.2 Perumusan Masalah

Dalam pelaksanaan penelitian ini, adapun masalah yang diangkat, dibahas, dan diselesaikan adalah:

1. Bagaimana menerapkan implementasi metode *LSB* pada penyembunyian pesan teks pada citra digital ?.
2. Bagaimana menganalisis kinerja dari metode *LSB* dilihat dari keberhasilan penyisipan dan ekstraksi pesan pada citra digital?

1.3 Batasan Masalah

Agar tidak memperluas materi penulisan maka batasan-batasan dan ruang lingkup penulisan antara lain adalah:

1. Format Citra yang digunakan berupa BMP, JPG dan PNG.
2. Ukuran Citra yang digunakan 100 x 100 sapai dengan 1024 x 768.
3. Pesan yang digunakan merupakan pesan teks (ASCII).

1.4 Tujuan Penelitian

Berdasarkan perumusan dan batasan masalah yang tertera diatas, ada pun tujuan dri penelitian ini, antara lain :

1. Untuk mengetahui mekanisme dari metode *LSB*..
2. Untuk meneliti metode *LSB* pada penyembunyian pesan teks pada citra digital.
3. Membangun aplikasi perangkat lunak komputer yang dapat digunakan untuk pengujian dan implementasi steganografi pesan teks pada citra digital.

1.5 Manfaat Penelitian

Dari penjabaran di atas ada pula manfaat yang diberikan, adalah :

1. Memahami bagaimana cara kerja metode *LSB* pada penyembunyian pesan teks pada citra digital.
2. Membantu pengguna dalam memahami dan menggunakan aplikasi steganografi *LSB*.

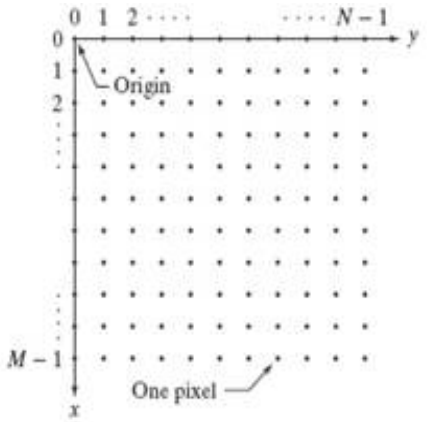
BAB II

LANDASAN TEORI

2.1 Citra Digital

Secara umum, citra digital merupakan gambar 2 dimensi yang disusun oleh data digital dalam bentuk sebuah larik (array) yang berisi nilai real maupun komplek yang direpresentasikan dengan deretan *bit* tertentu (Putra, 2010). Suatu citra dapat didefinisikan sebagai fungsi *f(x,y)* berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitude *f* di titik koordinat *(x,y)* dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut.

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau "*picture element*"). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan pada Gambar 1 berikut.



Gambar 2.1. Ilustrasi Citra Digital

(Sumber : Putra, 2010)

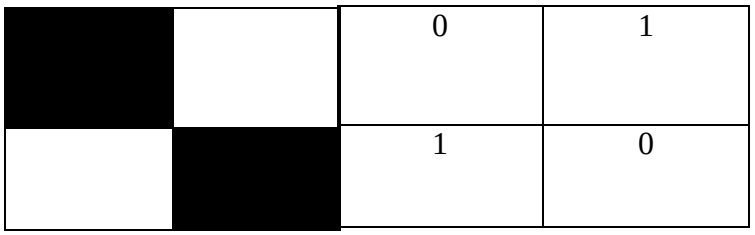
Dengan sistem koordinat yang mengikuti asas pemindaian pada layar TV standar itu, sebuah piksel mempunyai koordinat berupa (x, y) dalam hal ini:

1. x menyatakan posisi kolom;
2. y menyatakan posisi baris;
3. piksel pojok kiri-atas mempunyai koordinat (0, 0) dan piksel pada pojok kanan-bawah mempunyai koordinat (N-1, M-1).

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentuk. Format citra digital yang banyak dipakai adalah Citra Biner, Citra Grayscale, dan Citra Warna:

a. Citra Biner

Citra biner (*monochrome*) atau disebut juga *binary image*, merupakan citra digital yang setiap *pixel*-nya hanya memiliki 2 kemungkinan derajat keabuan, yaitu 0 dan 1 (Lusiana, 2013). Nilai 0 mewakili warna hitam, dan nilai 1 mewakili warna putih, di mana setiap *pixel*-nya membutuhkan media penyimpanan sebesar 1 *bit.*



Gambar 2.2. **Contoh Citra Biner Berukuran 2x2 *Pixel***

(Sumber : Lusiana, 2013)

b. Citra Warna

Setiap piksel pada citra warna memiliki warna yang merupakankombinasi dari tiga warna dasar *RGB* (*Red, Green, Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna (Adikara, 2014). Berarti setiap piksel mempunyai kombinasi warna sebanyak 28 . 28 . 28 = 224 = 16 juta warna lebih. Itulah yang menjadikan alasan format ini disebut dengan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bias dikatakan hampir mencakup semua warna di alam. Penyimpanan citra *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 byte. Sedangkan 1 piksel citra *true color* diwakili oleh 3 *byte*, dimana masing-masing *byte* merepresentasikan warna merah, hijau dan biru.

2.2 Pengolahan Citra Digital

1. Definisi Pengolahan Citra

Pengolahan citra adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images)* yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan

waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan output-nya adalah citra hasil pengolahan (Putra, 2010).

2. Tujuan Pengolahan Citra Digital

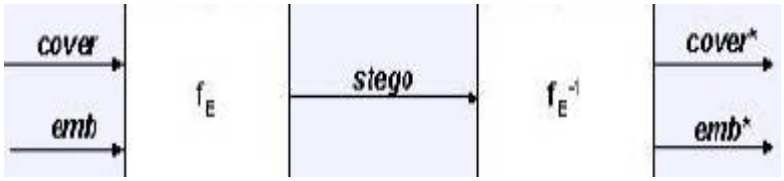
Pengolahan citra digital banyak dimanfaatkan oleh berbagai bidang mulai dari keamanan, kesehatan, pendidikan dan bidang – bidang yang lain. Berikut beberapa tujuan dari kegiatan pengolahan citra digital.

a. Memperbaiki kualitas gambar dilihat dari aspek *radiometric* (peningkatan kontras, tranformasi warna, restorasi citra) dan dari aspek *geometric* (rotasi, translasi, skala, transformasi geometrik).
b. Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.
c. Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data (Sutoyo, 2009).

2.3 Steganografi

Steganografi adalah ilmu dan seni dari komunikasi yang tidak terlihat (Wandani, 2012). Steganografi merupakan kata yang diturunkan dari kata-kata yunani yaitu "*stegos"* yang berarti "menutupi" dan "*grafia"* yang berarti menulis yang mana jika didefinisikan dapat dengan "tulisan yang ditutupi". Steganografi berbeda dari kriptografi dimana kriptografi bertujuan pada menjaga konten atau informasi dari pesan tetap rahasia sedangkan steganografi bertujuan untuk menjaga keberadaan pesan tetap rahasia.

Pesan asli disembunyikan pada sebuah media pembawa yang mana perubahan yang terjadi pada media pembawa tidak terlihat oleh orang lain (Kumar & Pooja, 2010). Kelebihan dari steganografi salah satunya adalah dimana pesan ditransmisikan atau dikirim tanpa diketahui oleh pihak lain yang mana bagi pihak lain yang terlihat adalah media pembawanya saja.

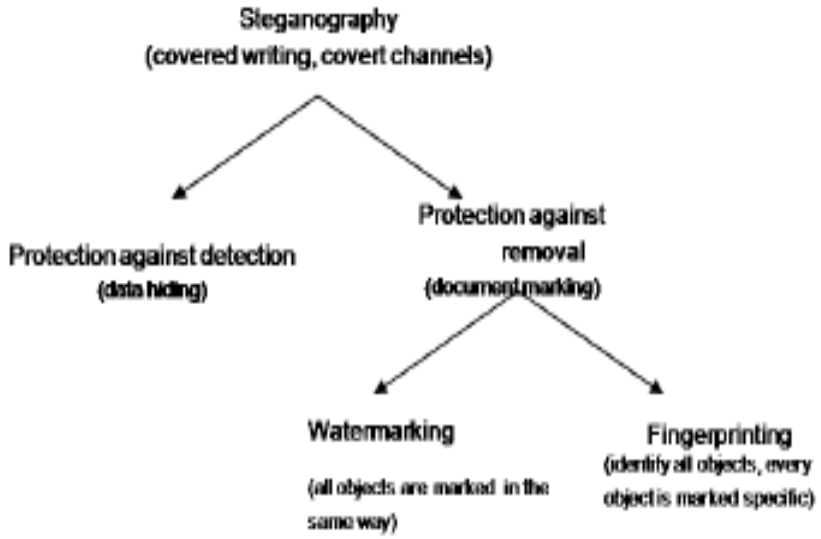


Gambar 2.3. Ilustrasi Sistem Steganografi.

(Sumber : Ginanjar, 2015)

Penggunaan steganografi adalah sebagai berikut :

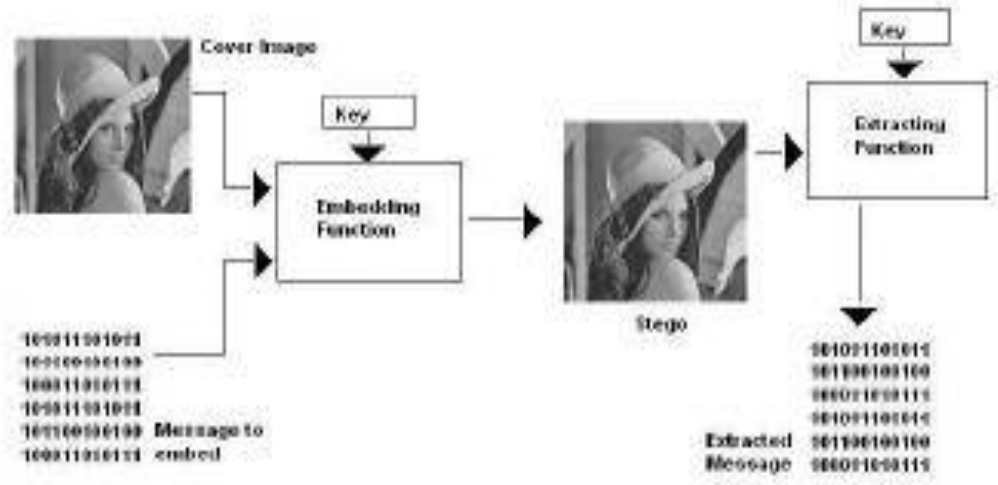
1. Steganografi dapat menjadi solusi yang mana memungkinkan untuk mengirim berita atau informasi dicegah oleh sensor atau khawatir terhadap pesan dibajak oleh pihak lain.
2. Steganografi juga dapat digunakan untuk menyimpan pada suatu lokasi seperti media digital lain.
3. Steganografi juga dapat digunakan sebagai watermarking pada media yang ingin dilindungi hak ciptanya.



Gambar 2.4. Tipe dari Steganografi.

(Sumber : Ginanjar, 2015)

Semua pendekatan yang ada pada bidang steganografi memiliki sebuah kesamaan yaitu menyembunyikan pesan rahasia pada objek fisik yang dikirimkan. Pada gambar diatas dapat dilihat proses dari steganografi dimana citra pembawa diteruskan kedalam fungsi penanaman yang kemudian akan menghasilkan citra yang telah mengandung pesan rahasia. Proses steganografi juga biasanya dapat menggunakan kunci untuk meningkatkan keamanan pada pesan yang disembunyikan, yang mana proses steganografi akan dilengkapi dengan proses kriptografi sebagai proses tambahan.



Gambar 2.5. Prosedur Steganografi.

(Sumber : Kumar & Pooja, 2010)

2.4 Steganografi *LSB*

LSB atau *Least Significant Bit* merupakan teknik yang umum digunakan pada bidang steganografi. Metode *LSB* bekerja dengan mengganti bit pada posisi *least significant* dengan bit dari informasi yang akan ditanam (Arifin, 2013). Berikut ilustrasi dari proses penanaman informasi menggunakan steganografi *LSB* pada media citra digital.

Proses *Embedding* :

Piksel : 00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

Karakter : A -> 65 -> 01000001

Hasil : (00100110 11101001 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

Proses *embedding* atau penanaman dilakukan dengan cara mengganti *bit LSB* pada citra dengan bit dari karakter informasi. *Bit* yang digaris bawah seperti yang terlihat pada proses *embedding* diatas merupakan bit pengganti yan diperoleh dari karakter informasi. Proses ekstraksi dilakukan dengan mengambil bit *LSB* dari tiap piksel dan kemudian merangkainya kembali menjadi karakter informasi.

Proses *Extracting* :

Hasil : (00100110 11101001 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

Ekstraksi Bit : 0 1 0 0 0 0 1 1

Desimal : 65

Karakter : A

2.5 Flowchart

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya (Nuraini, 2015). Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchar*t ini merupakan langkah awal pembuatan program.

Dengan adanya *flowchart* urutan proses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dan bahasa pemrograman.

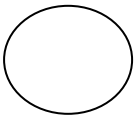
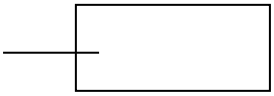
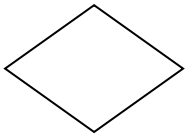
1. ***Flowchart* Sistem *(System Flowchart)***

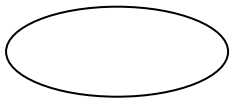
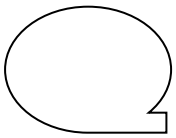
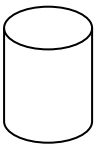
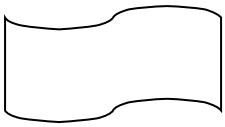
   Sistem *Flowchart* merupakan bagian yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Dengan kata lain, *flowchart* ini merupakan deskripsi secara frafik dari urutan prosedur-prosedur yang terkombinasi yang membentuk suatu sistem.

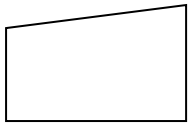
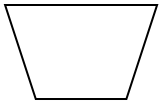
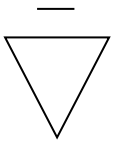
   Flowchart Sistem terdiri dari data yang mengalir melalui system dan proses yang mentransformasikan data itu. Data dan proses dalam *flowchart* sistem dapat digambarkan secara *online* (dihubungkan langsung dengan komputer) atau *offline* (tidak dihubungkan langsung dengan komputer, misalnya mesin tik, cash register atau kalkulator).

   Simbol-simbol yang digunakan dalam system *flowchart* antara lain :

Tabel 2.1. Simbol-simbol *flowchart*.

SIMBOL	NAMA SIMBOL / ARTI
	INPUT / OUTPUT Mempresentasikan input data atau output data yang diproses atau informasi
	PROSES Mempresentasikan operasi
	PENGHUBUNG Keluar atau masuk dari bagian lain flowchart khususnya halaman yang sama
	ANAK PANAH Mempresentasikan alur kerja
	PENJELASAN Digunakan untuk komentar tambahan
	KEPUTUSAN Keputusan dalam program
	PREDEFINED PROCESS Rincian operasi berada di tempat lain.
	PREPARATION Pemberian harga awal

	TERMILAN POINTS Awal / akhir flowchart
	PUNCHED CARD Input / output yang menggunakan kartu berulang
	DOKUMEN Input / output dalam format yang dicetak
	MAGNETIC TAPE Input / output yang menggunakan pita magnetic
	MAGNETIC DISK Input / Output yang menggunakan disk magnetic
	ON-LINA STORAGE Input / output yang menggunakan penyimpanan akses langsung
	PUNCHED TAPE Input / output yang menggunakan pita kertas berlubang

	MANUAL INPUT Input yang dimasukkan secara manual dari keyboard
	DISPLAY Output yang ditampilkan pada terminal
	MANUAL OPERATION Operasi manual
	OFF – LINE STORAGE Penyimpanan yang tidak dapat diakses oleh komputer secara langsung
	COMMUNICATION LINK Transmisi data melalui channel komunikasi, Seperti telepon

(Sumber : Blauch, 2012)

2.6 *UML(Unified Modeling Laguange)*

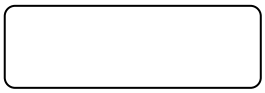
UML (Unified Modeling Laguange) adalah sebuah bahasa yang sudah menjadi standar dalam industry untuk merancang, menspesifikasi dan mendokumentasi sistem perangkat lunak (Hendini, 2016). Adapun tujuan dari *UML* adalah:

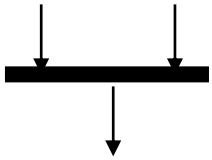
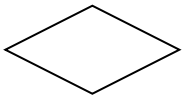
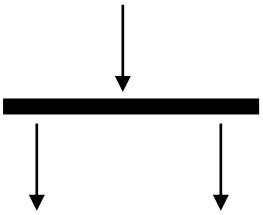
1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

2.7 *Activity*Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir (Hendini, 2016). Pada dasarnya, *activity* diagram merupakan variasi dari *statechart* diagram. *Activity* diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku paralel sedangkan *flowchart*tidak bisa. Berikut adalah notasi *activity* diagram.

Tabel 2.2. Notasi *Activity* Diagram

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	*Activity*

	*Connector*
	*Join*
	Decision Pilihan untuk mengambil keputusan
	Fork; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Note
	Receive Signal
	Send Signal
	Option Loop

Sumber : (Hendini, 2016)

2.8 *Sequence* Diagram

Sequence Diagram mendeskripsikan skenario (dapat mengacu pada expanded use case yang telah dibuat) dalam bentuk diagram (Nurdam, 2014). Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek – obyek yang melakukan suatu tugas atau aksi tertentu. Obyek – obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Pada diagram ini, dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis *Vertical*, disebut *lifeline*, dilekatkan pada setiap obyek atau aktor.

Tabel 2.3. Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Sumber: (Nurdam, 2014)

2.9 *Use Case* Diagram

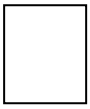
Use case diagram digunakan untuk menspesifikasikan fungsionalitas dari sistem (Kurniawan, 2018). *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

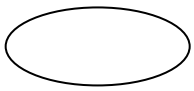
Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya.

Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *usecase* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel 2.4. Simbol *Use Case* Diagram

Gambar	Nama	Keterangan
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinterasi dengan *Use Case*.

	*Depedency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*Independent)*
	*Generalization*	Hubungan dimana objek anak(*Descended)* berbagi prilaku dan struktur data dari objek yang diatasnya objek induk.
	*Include*	Menspesifikasikan bahwa use case sumber secara explicit.
	*Extend*	Menspesifikasikan bahwa use case target memperluas prilaku pada use case sumber pada sebuah titik diberikan.
	Assosiation	Apa yang memhubungkan objek satu dengan objek yang lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur dari sebuah *actor.*
	*Colaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya.
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

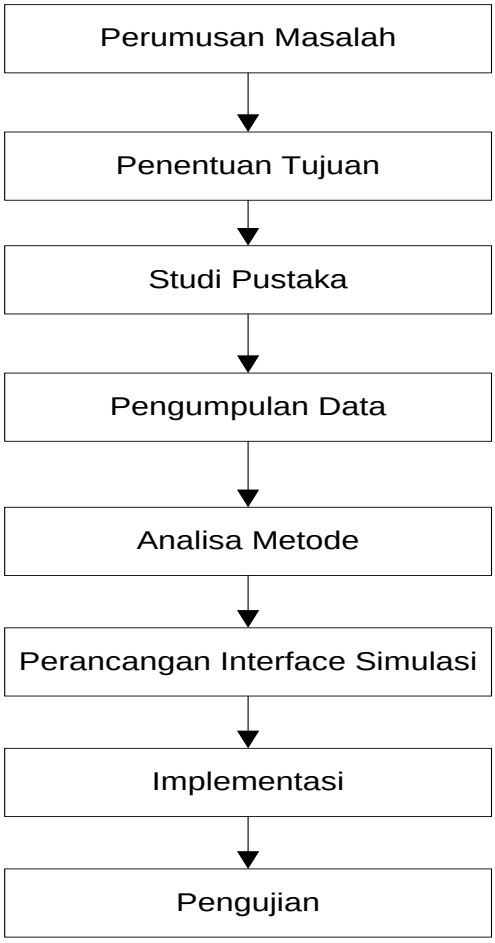
Sumber: (Kurniawan, 2018)

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Perancangan Sistem Aplikasi Keamanan Informasi Menggunakan Image Dengan Metode *LSB* adalah sebagai berikut:



Gambar 3.1 Tahapan Penelitian

3.2 Metode Pengumpulan Data

Pengumpulan data adalah pencarian terhadap susuatu kerena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 3, yaitu :

1. Pengamatan (*Observation*)

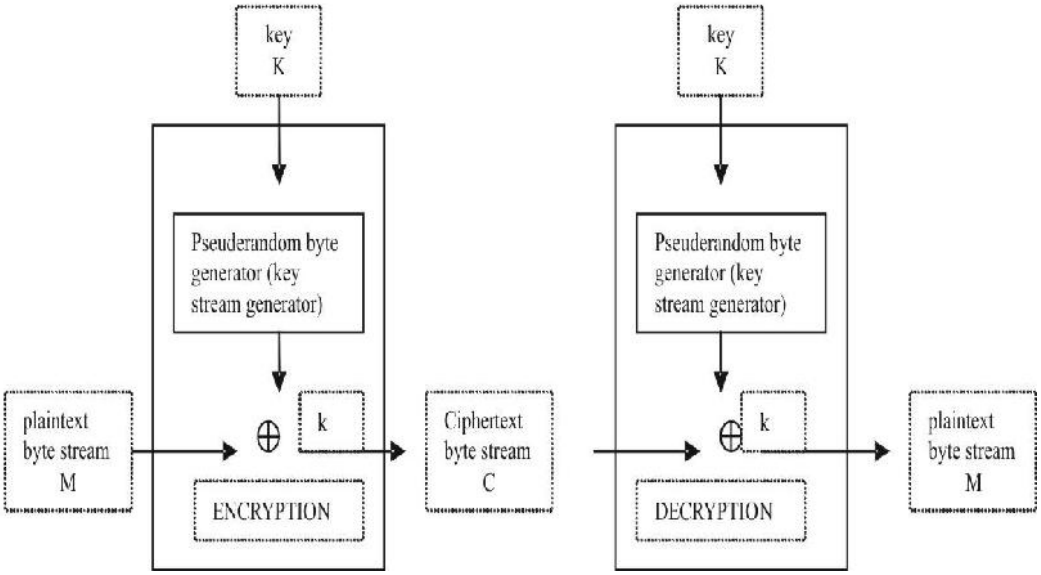
   Penulis melakukan pengamatan langsung pada setiap penggunan aplikasi chating yang sudah ada seperti *WA, BBM* dan *Line* untuk mengamati proses keamanan yang sudah dibuat sebelumnya.

2. Penelitian Kepustakaan (*Library Research*)

   Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus, maupun perpustakaan umum, juga melakukan pencarian lewat internet, dengan mengunjungi situs-situs seperti *google Book online* yang dapat membantu pembahasan materi.

3.3 Analisis Permasalahan yang Berjalan

Pertukaran data dalam hal ini pesan rahasia berbentuk teks dengan menggunakan metode tradisional yaitu dengan cara bertukar kata kunci tunggal. Diagram dibawah adalah penggambaran bagaimana pertukaran pesan rahasia menggunakan kunci tunggal terjadi.



Gambar 3.2 Analisis Permasalah yang Berjalan

Pemberitahuan kata kunci dari pengirim ke penerima menggunakan media yang umum digunakan oleh banyak orang.

3.4 Analisa Proses Sistem Yang Berjalan

Terdapat 2 (dua) proses utama dalam penyisipan pesan menggunakan metode *Least Significant Bit*, yaitu proses *embedding* dan proses *extraction.* Proses *embedding* adalah proses penyisipan pesan rahasia ke dalam suatu media. Sedangkan proses *extraction* adalah proses pengambilan pesan rahasia dari suatu media. Pada sistem ini, pesan rahasia yang digunakan berupa data*biner* teks yang merupakan *text* dari hasil enkripsi teknik steganografi ke dalam nilai bit akhir dari media penampung (Gambar *file*)dan media yang digunakan untuk penyisipan pesan adalah *file* Gambar berformat .bmp, .jpg.

1. Proses *Embedding*

Proses *embedding* atau penyisipan pesan menggunakan metode *Least Significant Bit* adalah sebagai berikut :

a) Inputkan Gambar yang akan menjadi media penyisipan *text* (*cover file*).

b) Inputkan *text* yang sudah terenkripsi untuk disisipkan.

c) Baca nilai *biner* setiap *pixel* Gambar.

d) Sisipkan nilai *biner* dari *text* pada nilai akhir *biner* dari *pixel* Gambar.

e) Petakan menjadi Gambar baru.

Berikut contoh penyisipan *text* menggunakan metode *Least Significant Bit*: Terdapat satu pesan yang sudah dienkripsi “AKU” yang akan disisipkan pada suatu Gambar.

Gambar 3.3. gambar.jpg

Langkah pertama adalah mengubah kedua data tersebut (kata AKU dan Gambar) menjadi biner.

Tabel 3.1. nilai biner teks AKU

Nilai Biner AKU		
A	K	U
0	0	0
1	1	1
0	0	0
0	0	1
0	1	0
0	0	1
0	1	0
1	1	1

Tabel 3.2. Tabel Biner Gambar

00000001	00010100	00000000	00000001	00010100	00000000	00000001	00010100
00000001	00000000	00010011	00000000	00000000	00010011	00000000	00000000
00010101	00000000	00000000	00010110	00000001	00000000	00011000	00000000
00000000	00011010	00000000	00000001	00010100	00000000	00000000	00010011

00000000	00000000	00010011	00000000	00000000	00010110	00000001	00000000
00010110	00000001	00000000	00010110	00000001	00000010	00010101	00000010
00000000	00010011	00000000	00000001	00010011	00000011	00000000	00010001
00000001	00000000	00010001	00000001	00000000	00010000	00000000	00000000

Kemudian gantikan tiap biner dari teksnya ke dalam akhir biner Gambar penampung, sehingga akan terlihat seperti pada tabel berikut ini.

Tabel 3.3. Tabel biner Gambar yang berisi pesan rahasia

00000**0**	00000**1**	00010**0**	00000**0**	00000**0**	00010**0**	00000**0**	0000000**1**	A
00010**0**	00000**1**	00000**0**	00010**1**	00000**1**	00000**0**	00011**0**	00000001	K
00000**0**	00011**1**	00000**0**	00000**1**	00011**0**	00000**1**	00000**0**	0001100**1**	U
00000000	00000000	00010101	00000000	00000000	00010011	00000000	00000000	-
00010011	00000000	00000000	00010111	00000001	00000000	00010111	00000001	-
000000	000101	000000	000000	000101	000000	000000	0001	-

00	11	01	10	01	10	00	0011	
00000000	00000000	00010011	00000011	00000000	00010001	00000001	00000000	-
00010001	00000001	00000000	00010001	00000000	00000000	00010001	00000000	-

Terlihat pada tiap akhir dari biner Gambar telah tersisipi oleh pesan rahasia yang ditandai dengan huruf *Bold* (cetak tebal). Langkah selanjutnya adalah matriks tersebut akan dipetakan kembali dalam bentuk Gambar dan Gambar ini disebut *stego file*.

2. Proses *Extraction*

Proses *extraction* atau pengambilan *text* dari media penampung menggunakan metode *Least Significant Bit* adalah sebagai berikut :

1. Masukkan Gambar yang telah disisipkan *text* (*stego file*).
2. Baca nilai biner dari *pixel stego file* yang terdapat pada biner terakhir *pixel* Gambar penampung.
3. Ambil nilai *binertext* yang terdapat pada *stego file*, yaitu nilai *biner* dari tiap-tiap *pixel* terakhir yang berubah.

Berikut contoh pengambilan *text* dengan menggunakan metode *Least Significant Bit*: Terdapat suatu Gambar “contoh.bmp” yang telah disisipkan *text* (*stego file*). Nilai setiap *pixel file* Gambar tersebut dapat dilihat pada Tabel 7.

Gambar 3.2. File Gambar

Kemudian *text* dibaca pada nilai akhir dari *biner pixel stego file* seperti pada tabel 7.

Tabel 3.4. Tabel biner Gambar yang berisi pesan rahasia

000000**0**	000000**1**	000100**1**	000000**0**	000000**0**	000100**1**	000000**0**	000000**1**	A
000101**0**	000000**1**	000000**0**	000101**1**	000000**1**	000000**0**	000110**1**	00000001	K
000000**0**	000110**1**	000000**0**	000000**1**	000110**0**	000000**1**	000000**0**	000110**1**	U
00000000	00000000	00010101	00000000	00000000	00010011	00000000	00000000	-
00010011	00000000	00000000	00010111	00000001	00000000	00010111	00000001	-

00000000	00010111	00000001	00000010	00010101	00000010	00000000	00010011	-
00000000	00000000	00010011	00000011	00000000	00010001	00000001	00000000	-
00010001	00000001	00000000	00010001	00000000	00000000	00010001	00000000	-

Dengan mengambil nilai biner *pixel* yang terakhir,yang dimulai dari awal pada baris pertama *pixel* Gambar, didapatlah nilai biner dari *text* yaitu "01000001=A, 01001011=K, 01010101=U".

Tabel 3.5.Tabel biner pesan rahasia yang disisipkan

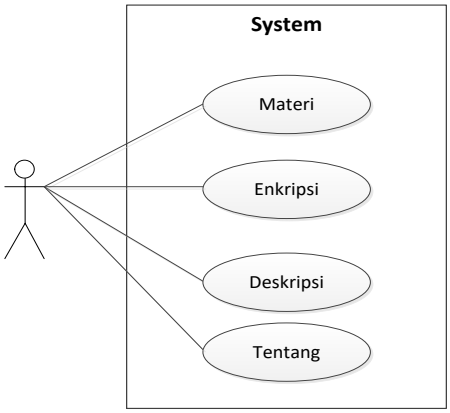
000000**0**	000000**1**	000100**1**	000000**0**	000000**0**	000100**1**	000000**0**	000000**1**	A
000101**0**	000000**1**	000000**0**	000101**1**	000000**1**	000000**0**	000110**1**	00000001	K
000000**0**	000110**1**	000000**0**	000000**1**	000110**0**	000000**1**	000000**0**	000110**1**	U

3.5 Perancangan Berorientasi Objek

Perancangan atau Pemodelan Berorientasi Objek merupakan proses mendapatkan informasi dari model dan menampilkannya secara grafik dengan menggunakan sebuah standar elemen grafik. Tujuan dari perancangan berorientasi ojbek ini memungkinkan adanya komunikasi yang lebih berkualitas antara pengguna, pengembang penganalisis, tetster, manajer dan siapapun yang terlibat dalam proyek pengembangan sistem informasi.

1. *Use case* Diagram

Berikut adalah *use case* diagram yang menggambarkan kegiatan.



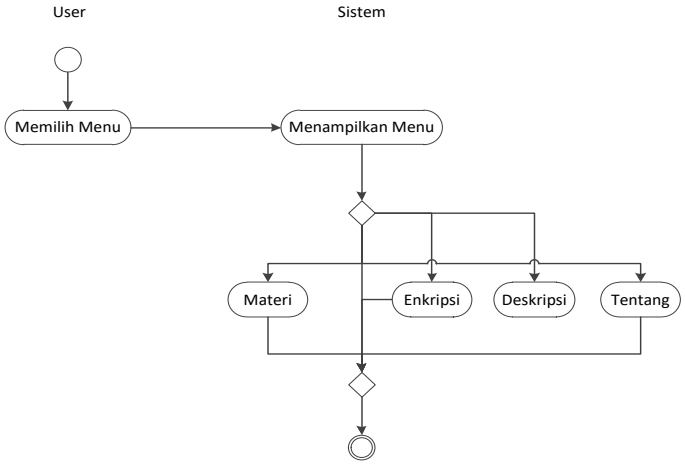
Gambar 3.6. *Use Case* Diagram

Keterangan :

Dalam use case diagram di atas, user/pengguna sebagai actor yang mempunyai use case Materi, Enkripsi dan Tentang.

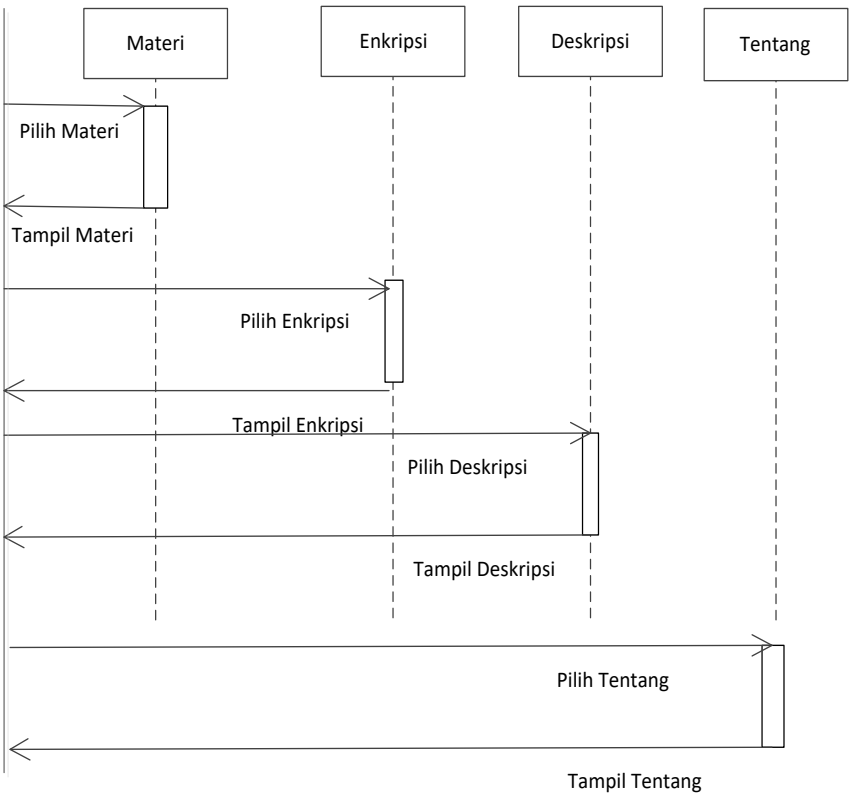
2. *Activity* Diagram

Activity diagram menggambarkan aktifitas-aktifitas yang terjadi dalam aplikasi dari aktivitas dimulai sampai aktivitas berhenti.



Gambar 3.*7 Activity* Diagram

3. *Sequence* Diagram



Gambar 3.8.*Sequence Diagram*

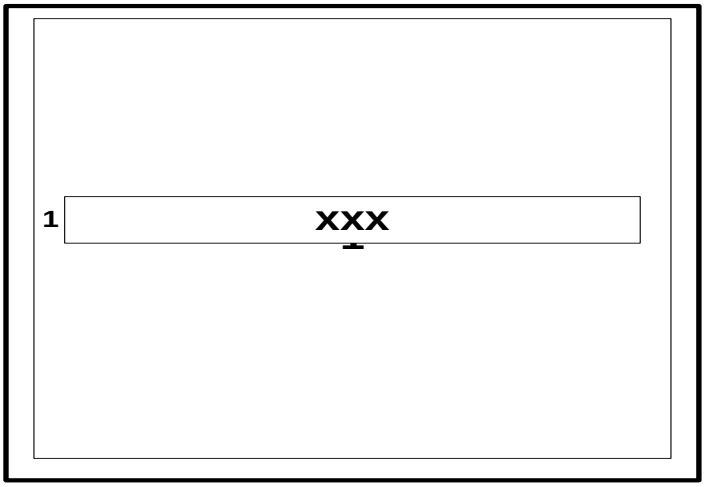
Keterangan Gambar :

1. Dalam diagram di atas menjelaskan bahwa *user* memilih materi kemudian Sistem menampilkan materi yang berkaitan dengan materi
2. *User merequest* Enkripsi kemudian Sistem menampilkan menu Enkripsi
3. *User merequest* Deskripsi kemudian Sistem menampilkan menu Deskripsi
4. *User merequest* Menu Tentang kemudian Sistem menampilkan *Form* Tentang.

3.6 Perancangan Antar muka

1. Rancangan Halaman Judul

Halaman judul merupakan halaman yang pertama muncul pada saat program dijalankan



Gambar 3.9 Rancangan Halaman Judul

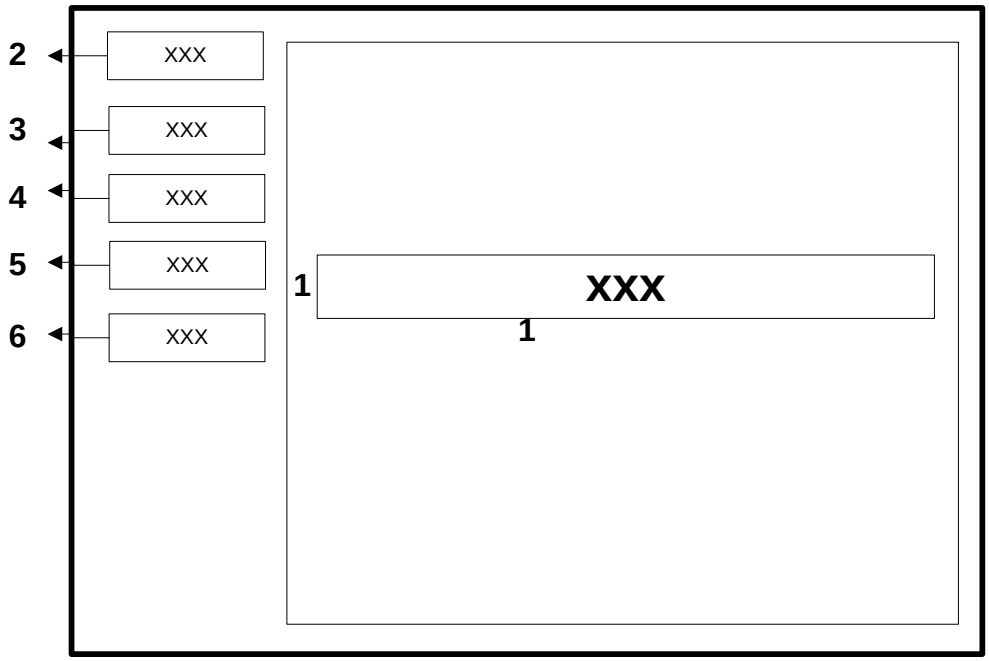
Pada rancangan di atas akan menampilkan judul yang kemudian akan pindah ke *form* menu utama dengan menggunakan *timer*.

Keterangan:

a. Berfungsi untuk menampilkan judul program.

2. Rancangan Halaman Menu Utama

Form ini berisi tombol-tombol seperti menu Materi, Enkripsi, Deskipsi, tentang, dan keluar.



Gambar 3.10 Rancangan Halaman Menu Utama

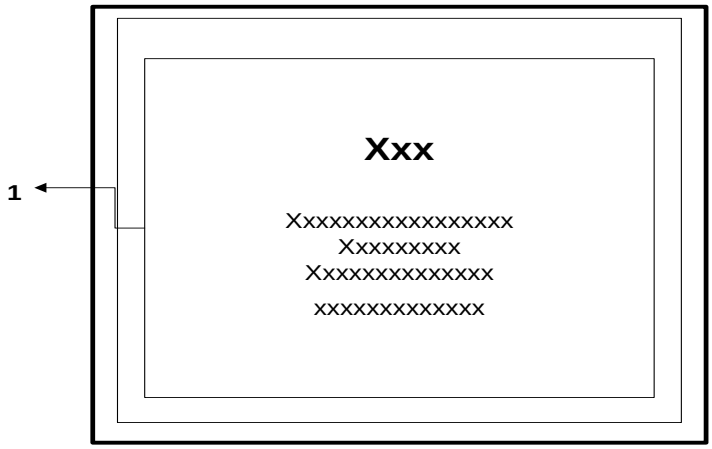
Pada tampilan di atas terdapat 5 tombol yaitu Materi, Enkripsi, Deskripsi, Tabel *Affine*, Tentang dan keluar.

Keterangan:

1. Tombol Berfungsi untuk menampilkan judul program.
2. Tombol Materi berfungsi untuk menghubungkan pengguna ke *form* materi.
3. Tombol Enkripsi berfungsi untuk menghubungkan pengguna ke *form* Enkripsi.
4. Tombol Deskripsi berfungsi untuk menampilkan form Deskripsi.
5. Tombol Tentang berfungsi untuk menghubungkan pengguna ke *form* tentang.
6. Tombol Keluar berfungsi untuk keluar dari program.

3. Rancangan Halaman Materi

Form ini digunakan untuk menjelaskan cara kerja penyandian, dimulai dari *plaintext* kemudian kunci yang dikonversikan dalam bentuk angka. Setelah itu dilakukan proses penjumlahan dan jika hasil penjumlahan maka akan dikurangi 6 lalu hasilnya akan dikembalikan lagi ke dalam bentuk huruf.



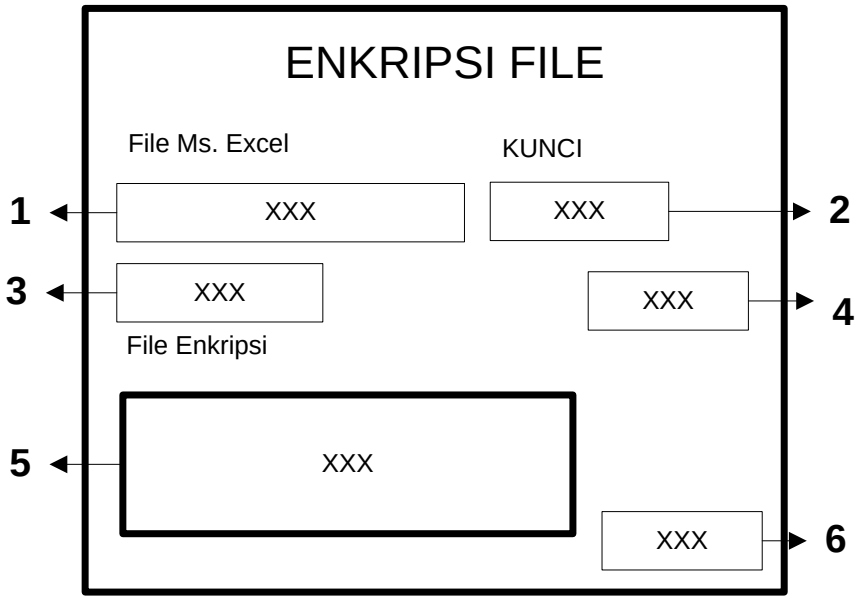
Gambar 3.11 Rancangan Halaman Materi

Keterangan:

1. Tombol Berfungsi untuk menampilkan Materi tentang Kriptografi *LSB*

3.7 Rancangan Halaman Enkripsi

Berisi penjelasan mengenai Enkripsi.Pengguna memasukkan tulisan asli atau *plaintext* ke dalam tombol masukan *plaintext* kemudian dimasukkan juga kunci. Setelah itu, ditekan tombol Proses Enkripsi yang kemudian akan menampilkan ciphertext atau tulisan yang telah disandikan.



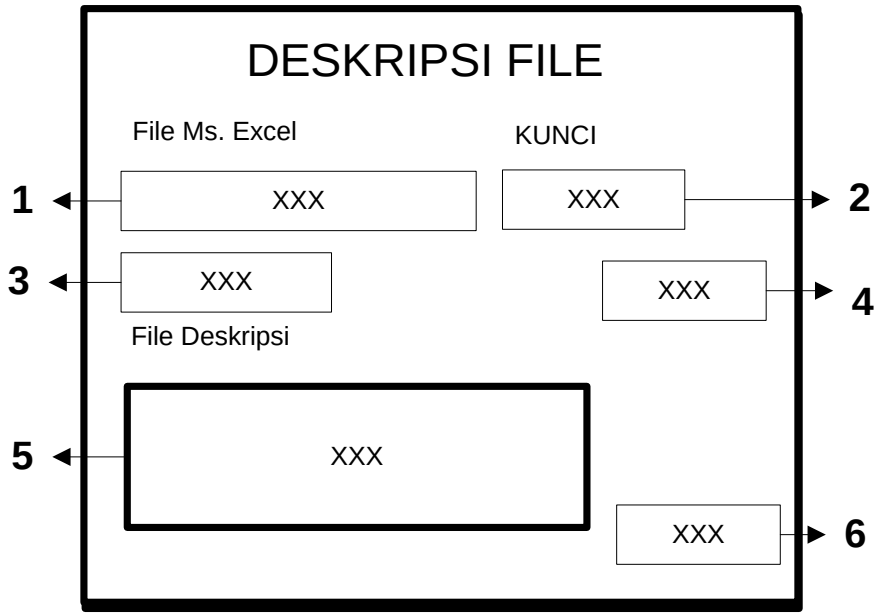
Gambar 3.12 Rancangan Halaman Enkripsi

Keterangan:

1. Berfungsi untuk menampilkan nama gambar yang sudah di *upload*.
2. Berfungsi untuk menginputkan kunci untuk mengenkripsi gambar.
3. Tombol yang berfungsi untuk mencari gambar yang ingin di enkripsi.
4. Tombol yang berfungsi untuk melakukan proses enkripsi pada gambar menggunakan *LSB.*
5. Berfungsi untuk menampilkan hasil dari proses enkripsi dari sebuah gambar.

3.8 Rancangan Halaman Deskripsi

Berisi penjelasan mengenai Enkripsi.Pengguna memasukkan tulisan asli atau *plaintext* ke dalam tombol masukan *plaintext* kemudian dimasukkan juga kunci. Setelah itu, ditekan tombol Proses Enkripsi yang kemudian akan menampilkan ciphertext atau tulisan yang telah disandikan.



Gambar 3.13 Rancangan Halaman Deskripsi

Keterangan:

1. Berfungsi untuk menampilkan nama gambar yang sudah di *upload*.
2. Berfungsi untuk menginputkan kunci untuk mengdeskripsi gambar.
3. Tombol yang berfungsi untuk mencari gambar yang ingin di deskripsi.
4. Tombol yang berfungsi untuk melakukan proses deskripsi pada Gambar menggunakan *LSB*.
5. Berfungsi untuk menampilkan hasil dari proses deskripsi dari sebuah gambar.

Pada gambar di atas terdapat kotak *input* Deskripsi berfungsi untuk memasukkan tulisan yang telah disandikan. Kemudian terdapat tombol Proses Deskripsi untuk mengembalikan ke tulisan asli jika kunci yang dimasukkan sama dengan kunci pada saat penggunaan *plaintext*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Algoritma

Algoritma adalah urutan langkah untuk menyelesaikan masalah secara sistematis dan logis. Algoritma menawarkan suatu metode dalam menyelesaikan sebuah permasalahan. Algoritma diartikan sebagai urutan langkah dalam menyelesaikan masalah secara sistematis dan logis. Pendekatan secara sistematis dan logis tersebut, menjadikan proses penyelesaian masalah terjaga kebenarannya karena algoritma haruslah benar agar dapat menghasilkan solusi yang benar.

4.2 Algoritma *Least Significant Bit* (*LSB*)

Algoritma *least significant bit* adalah algoritma yang di gunakan untuk menyisipkan data atau mengambil data dari dalam media penyimpanan yang digunakan. Algoritma steganografi *LSB* dibagi menjadi dua, yaitu menyisipkan data teks (*Embedded*) dan mengabil data teks (*Extraction*).

1. Proses Penyisipan Data Teks (*Embedded*)

Algoritma atau langkah-langkah untuk menyisipkan data teks pada data citra digital:

Input :C, T, KD, Pc, Pb, vM, vH, vB, toLSB, toDesimal, toBiner, xpix, Gp

Output : CT

Proses :

for Pc = 0 To panjang C -1

for Pb = 0 To panjang C -1

vM = C. Gp (Pb dan Pc) R

vH = C. Gp (Pb dan Pc) G

vB = C. Gp (Pb dan Pc) B

T1 = Mid i, 1

T2 = Mid i + 1, 1

T3 = Mid i + 2, 1

vM = toDecimal (toLSB (ToBiner (vM), T1))

vH = toDecimal (toLSB (ToBiner (vH), T2))

vB = toDecimal (toLSB (ToBiner (vB), T3))

xpix = xpix + 1

If xpix > xpx Then Exit For

i = i + 3

Next

CT ← Stego Image (gambar yang telah berisi pesan)

1. Proses Mengambil Data Teks (*Extraction*)

Algoritma atau langkah-langkah untuk membaca pesan pada data citra digital adalah sebagai berikut:

Input :SI, T, Pc, Pb, vM, vH, vB, toBiner, xpix, Gp, Gpes

Output : EP

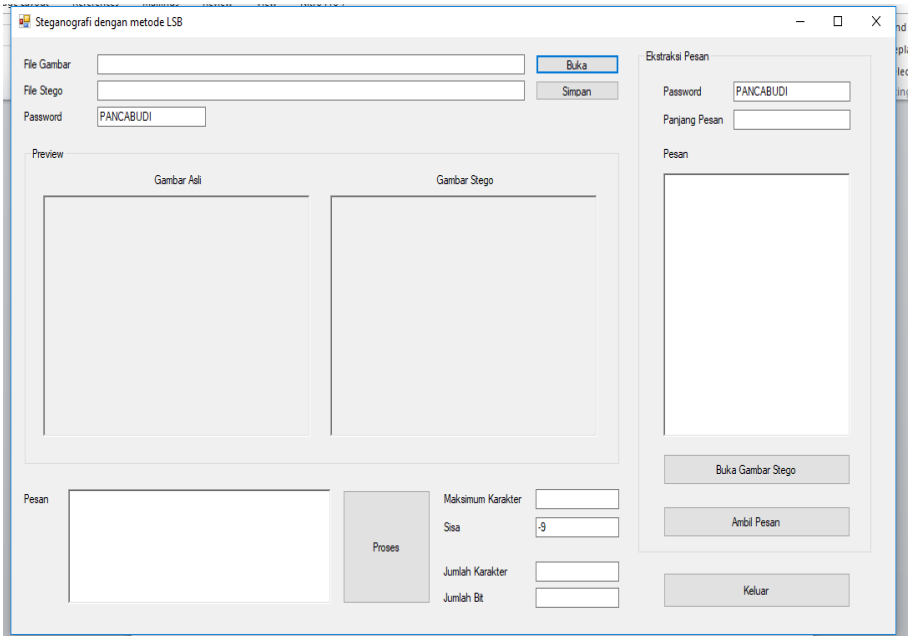
```
Proses :
        For Pc = 0 To SI.Height - 1
For Pb = 0 To SI.Height – 1
                vM = SI. Gp (Pb, Pc). R
                vH = SI .GP (Pb, Pc). G
                vB = SI. GP (Pb, Pc). B
        T. Mid ((ToBiner (vM)), 8, 1)
        T. Mid ((ToBiner (vH)), 8, 1)
        T. Mid ((ToBiner (vB)), 8, 1)
        xpix = xpix + 1
                If xpix > xpx Then Exit For
                Next
        T = T + 1 * 8
        Next
EP ← Pesan teks yang di ekstrak
```

4.3 Implementasi Sistem

Tahap implementasi merupakan lanjutan dari tahap perancangan sistem.Pada tahap ini dilakukan implementasi system kedalam bahasa pemrograman berdasarkan hasil analisa dan perancangan sistem.Pada tahap implementasi ini digunakan perangkat lunak dan perangkat keras, sehingga sistem yang dibangun dapat diselesaikan dengan baik.

4.4 Tampilan Halaman Steganografi

Halaman Steganografi merupakan halaman yang muncul pertama sekali pada saat system dijalankan. Tampilan halaman Steganografi dapat dilihat pada Gambar 15.



Gambar 4.1. Tampilan Halaman Menu Utama

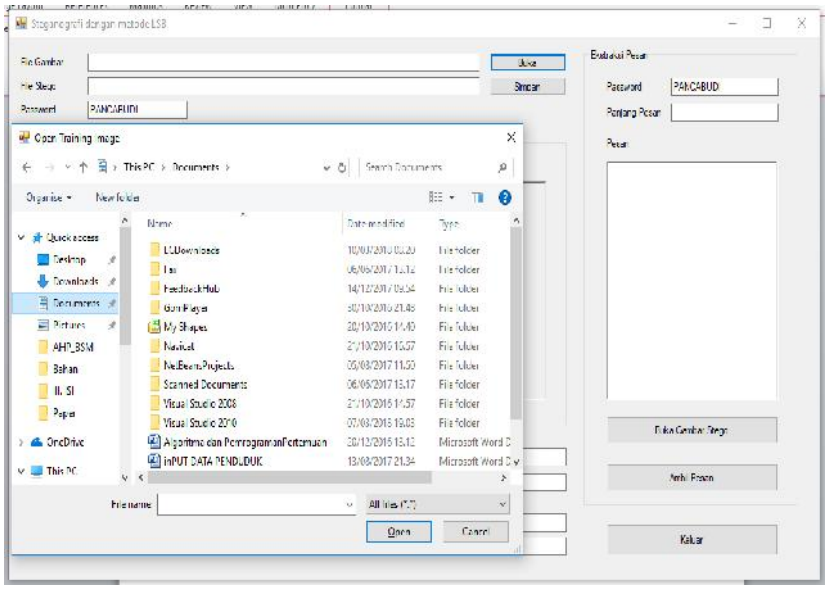
Keterangan:

1. Buka untuk mencari *file* gambar dengan format jpg, bmp, png.
2. Simpan untuk menyimpan gambar yang telah disisipkan pesan *text*.
3. *Picture Box* (Gambar Asli) untuk menampilkan gambar asli sebelum disisipkan pesan *text*.
4. *Picture Box* (Gambar Stego) untuk menampilkan gambar asli sesudah disisipkan pesan *text*.
5. *Text Box* (Pesan) untuk menginputkan pesan yang akan disisipkan ke *file* gambar.

6. *Button* (Proses) untuk memproses penyisipan pesan kedalam gambar dengan metode Steganografi.

4.5 Tampilan Cari Gambar

Halaman Cari Gambar merupakan halaman yang muncul pada saat proses untuk penyembunyian pesan. Tampilan halaman Cari Gambar dapat dilihat pada Gambar 16.



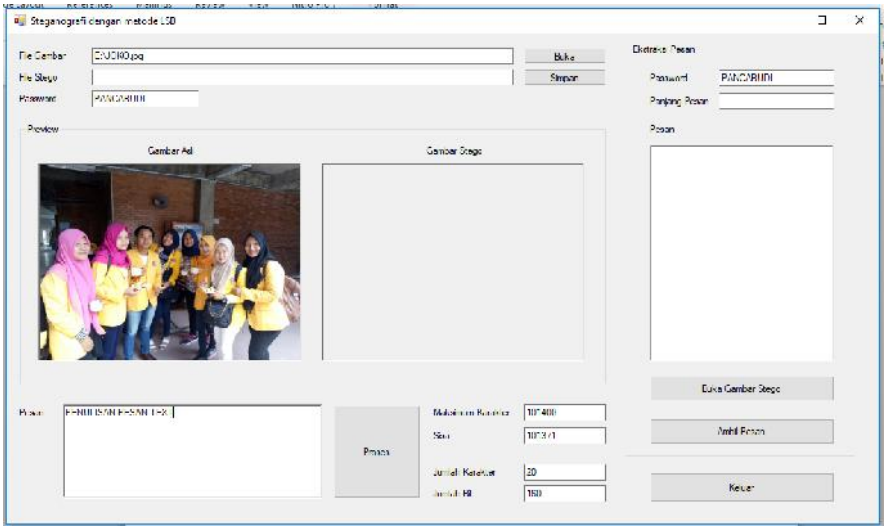
Gambar 4.2. Tampilan Cari Gambar

Keterangan:

1. Klik Buka pada *Text Box File* Gambar
2. Lalu, pilih gambar yang akan disisipkan oleh pesan *text*, dengan *file* gambar bertipe JPGE.
3. Lalu Klik OK.

4.6 Tampilan Penyembunyian Pesan *Text*

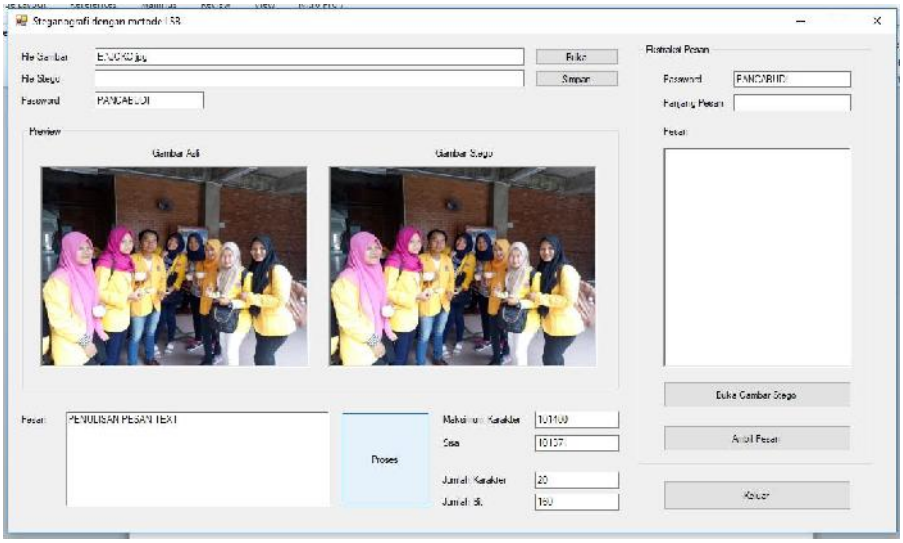
Halaman Penyembunyian Pesan *Text* merupakan halaman yang muncul pada saat proses untuk penyembunyian pesan. Tampilan halaman Penyembunyian Pesan *Text* dapat dilihat pada Gambar 17.



Gambar 4.3. Tampilan Penyembunyian Pesan *Text*

Keterangan:

1. Klik Buka pada *Text Box* File Gambar
2. Lalu, pilih gambar yang akan disisipkan oleh pesan *text*, dengan *file* gambar bertipe JPGE.
3. Lalu Klik OK.
4. Setelah muncul file gambar pada *Picture Box* (Gambar Asli), maka ketikkan pesan pada *Text Box* Pesan.
5. Lalu Proses.
6. File Gambar pada *Picture Box* (Gambar Asli) akan membuat file gambar baru yang muncul pada *Picture Box* (Gambar Stego).



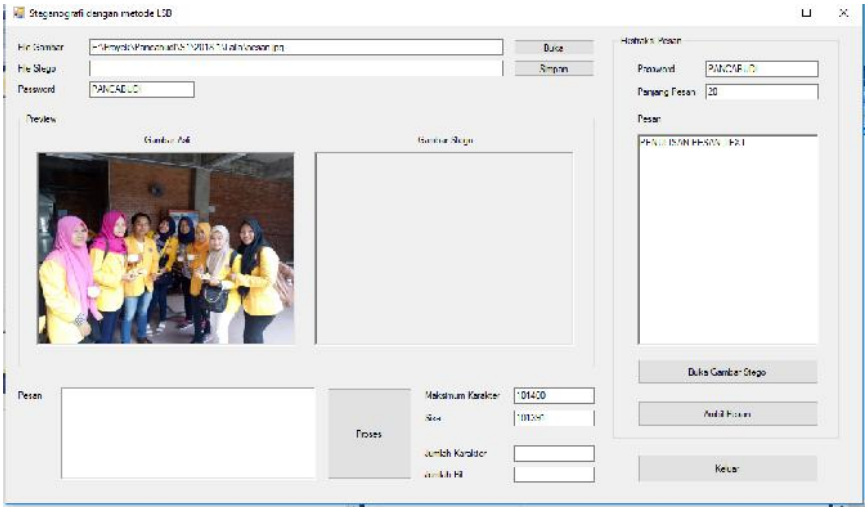
Gambar 4.4. Tampilan Gambar Yang Tersimpan *Text*

Keterangan:

1. Klik Buka pada *Button* "Buka Gambar *Stego*"
2. Lalu pilih gambar untuk melihat pesan.
3. Setelah gambar muncul pada *Picture Box* (Gambar Asli), maka masukkan *password* sesuai dengan pada saat proses penyisipan pesan.
4. Setelah *password* di inputkan, maka klik *Button* "Ambil Pesan".
5. Maka pesan yang tersimpan pada gambar, akan muncul pada *Text Box* pesan.
6. Selesai.

a. Tampilan Menampilkan Pesan *Text*

Halaman Menampilkan Pesan *Text* merupakan halaman yang muncul pada saat proses untuk menampilkan pesan. Tampilan halaman menampilkan Pesan *Text* dapat dilihat pada Gambar 19.



Gambar 4.5. Tampilan Penyembunyian Pesan *Text*

Keterangan:

1. Klik Buka pada *Button* File Buka Gambar *Stego*
2. Lalu, pilih gambar yang telah disisipkan oleh pesan *text*, dengan file gambar bertipe JPGE.
3. Lalu Klik OK.
4. Setelah muncul file gambar pada *Picture Box* (Gambar Asli), maka klik *button* ‘Ambil Pesan’.
5. Lalu Proses.
6. Maka pesan yang tadinya tersisipkan pada gambar, muncul pada *textbox* pesan.

4.7 Pengujian Sistem

Perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, perancangan, dan pengkodean.Pengujian yang digunakan untuk menguji system ini adalah metode pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak.

4.8 Rencana Pengujian

Pengujian fungsi Implementasi Steganografi *Lsb* Pada Penyembunyian Pesan Teks Pada Citra Digital ini dilakukan dengan menggunakan metode *Black Box*. Pengujian dilakukan pada fungsi-fungsi system untuk menentukan apakah fungsi tersebut telah berjalan sesuai dengan yang diharapkan.

1) Rencana Pengujian Cari Gambar

Tabel 4.1 . Rencana Pengujian Cari Gambar

Menu yang diuji	**Detail pengujian**	**Jenisuji**
Menu Utama	Tampilan Halaman Awal	*Black box*
Mengelola proses penyembunyian pesan *text*	*Input* Gambar	*Black box*
	Input Pesan	*Black box*
	Input Password	*Black box*

2) Rencana Pengujian Pengujian Pengguna

Tabel 4.2. Rencana Pengujian Pengguna (*User*)

Menu yang diuji	**Detail pengujian**	**Jenis uji**
Input Password	Menginputkan Key Pada Pesan	*Black box*
Input Gambar	Mencari Gambar Untuk Media Pesan	*Black box*
Input Pesan	Menampilkan Pesan yang ada pada Gambar.	*Black box*

Rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :

1) *Input* Gambar

Tombol cari gambar diuji untuk melihat efektifitas dari *button* tersebut, apakah *button* berfungsi dengan baik. Hasil uji dapat dilihat pada table berikut :

Table 4.3. Pengujian *Input* Gambar

Nama fungsi	Buka (File Gambar)
Tujuan	Untuk menguji *link* berfungsi dengan baik
Aktor	Pengguna (*user*)
Kondisi awal	Berada dihalaman utama
Kondisi akhir	File Gambar Muncul Pada *Picture Box*
Skenario	1) Aktor menekan *Button* Buka, dengan *Text Box* File Gambar 2) Sistem akan memunculkan Tampilan *Explore*

	Windows untuk mencari gambar yang ada pada PC atau Komputer 3) Jika sudah menemukan gambar, klik OK. Maka gambar akan masuk kedalam sistem.
Hasil yang didapat	Gambar Muncul pada Sistem
Kesimpulan	Fungsi berjalan dengan baik

2) *Input* Pesan

Tombol *input* pesan diuji untuk melihat efektifitas dari *button* tersebut, apakah *button* berfungsi dengan baik. Hasil uji dapat dilihat pada table berikut :

Tabel 4.4. Pengujian *Input* Pesan

Nama fungsi	Proses (*Button*)
Tujuan	Untuk menguji apakah proses tersebut sesuai dengan yang diinginkan
Aktor	Pengguna (*user*)
Kondisi awal	Berada pada Menu Utama
Kondisi akhir	Menghasilkan Pesan pada gambar yang sudah tersisipkan *text.*
Skenario	1) Aktor menginputkan pesan *text* pada *text box* proses 2) Sistem akan menyisipkan pesan tersebut kedalam gambar, dan akan menampilkan gambar tersebut di *Picture Box* Steganografi. 3) Lalu, klik simpan untuk menyimpan gambar yang telah di

	sisipkan *text*.
Hasil yang didapat	Gambar yang telah disisipkan Pesan (*Button* Simpan)
Kesimpulan	Fungsi berjalan dengan baik

3) Input Password

Tombol *input password* diuji untuk melihat efektifitas dari *text box*t ersebut, apakah *text box* berfungsi dengan baik. Hasil uji dapat dilihat pada table berikut :

Tabel 4.5. Pengujian *Input Passoword*

Nama fungsi	*Text Box (Password)*
Tujuan	Untuk menguji apakah proses tersebut sesuai dengan yang diinginkan
Aktor	Pengguna (*user*)
Kondisi awal	Berada pada Menu Utama
Kondisi akhir	Menghasilkan *Password* pada gambar yang sudah tersisipkan text untuk keamanan pesan.
Skenario	1. Aktor menginputkan *Password* pada *text box Password* 2. Sistem akan memberikan keamanan tersebut kedalam gambar, dan meminta konfirmasi *password* saat akan menampilkan gambar tersebut di *Picture Box* Steganografi.
Hasil yang didapat	*Password* pada gambar (*Button* Simpan)
Kesimpulan	Fungsi berjalan dengan baik

4) Menampilkan Pesan

Menampilkan Pesan diuji untuk melihat efektifitas dari *text box*tersebut, apakah *text box* berfungsi dengan baik. Hasil uji dapat dilihat pada table berikut :

Tabel 4.6. Pengujian Menampilkan Pesan

Nama fungsi	Buka Gambar *Stegano*
Tujuan	Untuk menguji apakah proses tersebut sesuai dengan yang diinginkan
Aktor	Pengguna (*user*)
Kondisi awal	Berada pada Menu Utama
Kondisi akhir	Menghasilkan pesan *text* yang dihasilkan dari gambar *stego*.
Skenario	1. Aktor mengklik *button* 'Ambil Gambar *Stego*', 2. Lalu, masukkan *password* pada *text box password*. 3. Setelah itu, klik *button* 'Ambil Pesan', jika sesuai *password* dengan gambar, maka pesan akan muncul pada *text box* pesan.
Hasil yang didapat	Pesan *text* pada *text box stego*.
Kesimpulan	Fungsi berjalan dengan baik

4.9 Kesimpulan dan hasil pengujian alpha

Hasil pengujian dari pengujian sistem telah selesai, menunjukkan bahwa system sudah memenuhi syarat fungsional. Secara fungsional sistem yang sudah dibangun sudah dapat menghasilkan keluaran sesuai yang diharapkan.

Tabel 4.7. Kesimpulan Pengujian Sistem

Nama fungsi	**Hasil**
Password	Fungsi berjalan dengan baik
Menampilkan Pesan	Fungsi berjalan dengan baik
Input Gambar	Fungsi berjalan dengan baik
Input Pesan	Fungsi berjalan dengan baik
Input Password	Fungsi berjalan dengan baik

BAB V

PENUTUP

5.1 Kesimpulan

Setelah keseluruhan proses dilakukan, yaitu dimulai dari tahapan studi literatur hingga pengujian perangkat lunak, maka dapat diambil kesimpulan sebagai berikut:

1. Algoritma steganografi *Least Significant Bit* dilakukan dengan menggantikan *bit-bit* pesan rahasia pada *bit* terakhir tiap komponen warna piksel citra. Satu komponen warna citra hanya disisipkan satu *bit* pesan (bernilai 0 atau 1) sehingga ukuran citra tidak berubah.
2. Kecepatan waktu proses bergantung pada besarnya file, panjang kunci dan kecepatan prosessor komputer yang digunakan.

5.2 Saran

Adapun saran-saran yang dapat penulis berikan untuk pengembangan dan perbaikan sistem ini adalah sebagai berikut :

1. Penelitian ini dapat dikembangkan dengan mencoba menerapkan beberapa metode lainnya seperti (algoritma *RSA* dan *DES*) sehingga pendeteksian pesan tersembunyi pada sebuah gambar lebih akurat dan sulit untuk dipecahkan.
2. Pada proses penyembunyian pesan sebaiknya dikombinasi dengan metode lainnya agar pesan yang disisipkan pada gambar menjadi lebih aman.

DAFTAR PUSTAKA

Adikara, P. P., Rahman, M. A., & Santosa, E. (2014). Pencarian Ruang Warna Kulit Manusia Berdasarkan Nilai Karakteristik (λ) Matrik Window Citra. Jurnal Teknologi Informasi dan Ilmu Komputer, 1(1), 29-33.

Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." Seminar Nasional Informatika (SNIf). Vol. 1. No. 1. 2017.

Arifin, R., & Oktoviana, L. T. (2013). Implementasi Kriptografi dan Steganografi menggunakan Algoritma RSA dan metode LSB. Universitas Malang.

Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.

Fachri, Barany. Aplikasi Perbaikan Citra Efek Noise Salt & Papper Menggunakan Metode Contraharmonic Mean Filter. In: Seminar Nasional Royal (Senar). 2018. P. 87-92.

Ginanjar, S. A., & Sugiharto, A. (2015). Steganografi Pesan Suara Ke Dalam Citra Menggunakan Persamaan Lingkaran Dan Metode Least Significant Bit (Doctoral dissertation, Universitas Diponegoro).

Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. Int. J. Recent Trends Eng. Res, 3(8), 58-64.

Hafni, Layla, And Rismawati Rismawati. "Analisis Faktor-Faktor Internal Yang Mempengaruhi Nilai Perusahaan Pada Perusahaan Manufaktur Yang Terdaftar Di Bei 2011-2015." Bilancia: Jurnal Ilmiah Akuntansi 1.3 (2017): 371-382.

Hamdi, Muhammad Nurul, Evi Nurjanah, And Latifah Safitri Handayani. "Community Development Based Onibnu Khaldun Thought, Sebuah Interpretasi Program Pemberdayaan Umkm Di Bank Zakat El-Zawa." El Muhasaba: Jurnal Akuntansi (E-Journal) 5.2 (2014): 158-180.

Hendini, A. (2016). Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). Jurnal Khatulistiwa Informatika, 4(2).

Indra Permana, Aminuddin "Sistem Pakar Mendeteksi Hama Dan Penyakit Tanaman Kelapa Sawit Pada Pt. Moeis Kebun Sipare-Pare Kabupaten Batubara." (2013).

International Journal of Computer Applications, 9(7), 19-23.

Kumar, A., & Pooja, K. (2010). Steganography-A data hiding technique.

Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. Jurnal Teknologi Informasi dan Ilmu Komputer, 5(1), 77.

Lusiana, V. (2013). Deteksi Tepi pada Citra Digital menggunakan Metode Kirsch dan Robinson. Dinamik, 18(2).

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Nuraini, R. (2015). Desain Algorithma Operasi Perkalian Matriks Menggunakan Metode Flowchart. Jurnal Teknik Komputer, 1(1), 144-151.

Nurdam, N. (2014). Sequence Diagram sebagai perkakas perancangan antarmuka pemakai. ULTIMATICS, 6(1), 21-25.

Permana, A. I., and Z. Tulus. "Combination of One Time Pad Cryptography Algorithm with Generate Random Keys and Vigenere Cipher with EM2B KEY." (2020).

Permana, Aminuddin Indra. "Kombinasi Algoritma Kriptografi One Time Pad dengan Generate Random Keys dan Vigenere Cipher dengan Kunci EM2B." (2019).

Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN. 2015.

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Putra, I. K. G. D., & Suarjana, I. G. (2010). Segmentasi citra retina digital retinopati diabetes untuk membantu pendeteksian mikroaneurisma. Majalah Ilmiah Teknologi Elektro.

Rizal, Chairul. "Pengaruh Varietas dan Pupuk Petroganik Terhadap Pertumbuhan, Produksi dan Viabilitas Benih Jagung (Zea mays L.)." ETD Unsyiah (2013).

Sutoyo, T. D., Mulyanto, E., & Suhartono, V. (2009). Teori Pengolahan Citra Digital.

Syahputra, Rizki, And Hafni Hafni. "Analisis Kinerja Jaringan Switching Clos Tanpa Buffer." Journal Of Science And Social Research 1.2 (2018): 109-115.

Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." Jurnal Abdi Ilmu 10.2 (2018): 1899-1902.

Wandani, H., Budiman, M. A., & Sharif, A. (2012). Implementasi sistem keamanan data dengan menggunakan teknik steganografi end of file (EOF) dan Rabin public key cryptosystem. Alkhawarizmi, 1(1).