PENERAPAN ALGORITMA LSB (LEAST SIGNIFICANT BIT) UNTUK PENYEMBUNYIAN TEKS PADA FILE IMAGE

Disusun dan Diajukan Sebagai Salah Satu Syarat untuk Menempuh Ujian Akhir
Memperolah Gelar Sarjana Komputer Pada Fakultas Sains Dan Teknologi
Universitas Pembangunan Panca Budi Medan

SKRIPSI

OLEH

NAMA	**: FREDY SUHENDY**
N.P.M	**: 1514370263**
PROGRAM STUDI	**: SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

FREDY SUHENDY

Penerapan Algoritma Lsb (Least Significant Bit) Untuk Penyembunyian Teks Pada File Image

2020

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan sebuah pesan. Dengan steganografi sebuah pesan tertulis dapat disembunyika dengan tinta yang tidak terlihat diantara garis-garis yang terlihat. Teknik steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks) didalam sebuah berkas seperti gambar. Tujuan dari steganografi adalah untuk menyembunyikan atau merahasiakan keberadaan sautu pesan. Dalam penelitian ini digunakan steganografi dengan metode *Least Significant Bit*, dimana media yang digunakan sebagai wadah untuk penyembunyian pesan tersebut berupa citra (gambar). Metode ini bekerja dengan cara menyisipkan ke dalam bit terendah pada data *pixel* yang menyusun file gambar BMP 24 bit. Pada file gambar BMP 24 bit setiap *pixel* pada gambar terdiri dari susunan tiga warna yaitu merah, hijau, biru (RGB).

Kata Kunci: *Steganografi, Least, Significant, Bit*

ABSTRACT

FREDY SUHENDY

Penerapan Algoritma Lsb (Least Significant Bit) Untuk Penyembunyian Teks Pada File Image

2020

Steganography is the art and science of writing hidden messages or hiding a message. With steganography a written message can be hidden with ink that is not visible between visible lines. Steganography techniques include many communication methods to hide secret messages (text) in a file like the picture. The purpose of steganography is to conceal or conceal the existence of a message. In this study used steganography with the *Least Significant Bit* method, where the media used as a container for hiding the message in the form of images (images). This method works by inserting into the lowest bit on the pixel data that composes a 24-bit BMP image file. In the 24 bit BMP image file each pixel in the image consists of three colors, red, green, blue (RGB).

Keywords: *Steganography*, *Least*, *Significant*, *Bit*

DAFTAR ISI

DAFTAR GAMBAR

Gambar **Halaman**

DAFTAR TABEL

Tabel **Halaman**

KATA PENGANTAR

Alhamdulillah, dengan mengucapkan puji dan syukur kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi ini. Shalawat beriring salam dipanjatkan untuk junjungan kita Nabi Muhammad SAW yang telah mengantarkan umatnya dari alam kegelapan dan kebodohan menuju alam yang terang benderang dan penuh dengan ilmu pengetahuan.

Selesainya penulisan skripsi ini telah banyak dibantu oleh berbagai pihak. Pada kesempatan ini, penulis ingin mengucapkan banyak terima kasih kepada pihak-pihak yang telah banyak membantu dan mendoakan penulis dalam menyelesaikan skripsi ini.

Adapun pihak-pihak yang telah banyak membantu penulis dan dengan kerendahan hati penulis mengucapkan terima kasih kepada :

1. Kepada kedua orang tua Dedy Suhendy dan Ibu Asnani yang sangat saya sayangi yang telah memberikan dukungan dan doa penuh sehingga skripsi ini selesai.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E., M.M. selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Bapak Hamdani, S.T., M.T. Selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Eko Hariyanto, S.Kom., M.Kom. selaku Ketua Program Studi Sistem Komputer.
5. Ibu Leni Marlina, S.Kom., M.Kom. selaku Dosen Pembimbing pertama penulis, yang sudah banyak membantu penulis dalam menyelesaikan skripsi ini dengan baik.
6. Bapak Rian Farta Wijaya, S.Kom., M.Kom. selaku Dosen Pembimbing kedua penulis, yang telah banyak memberi bimbingan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
7. Chandra Siswoyo, Chandra Prayoga, Saraswati, Regita Afrilla, Agun Nok yang selalu menemani, membantu dan menyemangati penulis dalam menyelesaikan skripsi ini sampai selesai.

8. Dan semua teman-teman yang ikut serta dalam mendoakan dan membantu penulisan skripsi ini, yang tak mungkin diucapkan satu persatu.

Penulis juga menyadari bahwa dalam penulisan skripsi ini, penulis memiliki banyak kekurangan dan memiliki kemampuan yang terbatas,sebab itu dengan segala kerendahan hati, penulis mengharapkan kritik dan saran yang membangundari berbagai pihak demi kesempurnaan penulisan skripsi ini.

Semoga Allah SWT, yang akan membalas segala kebaikan ini, dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Medan, Oktober 2019

(FREDY SUHENDY)
1 5 1 4 3 7 0 2 6 3

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi semakin memudahkan penggunanya dalamberkomunikasi melalui bermacam-macam media. Komunikasi yang melibatkanpengiriman dan penerimaan pesan dengan memanfaatkan kemajuan teknologiinformasi rentan terhadap pelaku kejahatan komputer yang memanfatkan celahkeamanan untuk mendeteksi dan memanipulasi pesan.

Keamanan dan kerahasiaan menjadi aspek yang sangat penting bagi pengguna teknologi informasi. Untuk menghindari pesan yang dikirimkan jatuh pada pihak-pihak yang tidak berkepentingan dan terjadi penyalahgunaan terhadap pesan, maka dilakukan enkripsi terhadap pesan asli dan penyisipan pesan ke dalam suatu media dengan menerapkan ilmu steganografi.

Untuk meningkatkan keamanan digunakan steganografi, dimana suatu sistem steganografi sedemikian rupa menyembunyikan isi suatu informasi di dalam suatu media yang tidak dapat di duga oleh orang biasa sehingga tidak membangunkan suatu kecurigaan kepada orang yang melihatnya.Media untuk menyembunyikan informasi adalah Format *image* jpeg, Png. Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah informasi.

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.

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* (Singh & Singh, 2015).

Berdasarkan paparan diatas, penulis ingin membuat skripsi dengan judul **"PENERAPAN ALGORITMA LSB (LEAST SIGNIFICANT BIT) UNTUK PENYEMBUNYIAN TEKS PADA FILE IMAGE"**

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, adapun rumusan masalah yang akan dibahas penulis 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

Berdasarkan perumusan masalah diatas maka penulis melakukan pembatasan masalah yang akan dibahas sebagai berikut:

1. Implementasi enkripsi dan dekripsi menggunakan file imagedengan algoritma LSB (Least Significant Bit).
2. Program yang dibahas menggunakan pemrograman Visual Basic.Net 2019.
3. Penerapan enkripsi hanya bisa menggunakan 255 karakter.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian dengan menggunakan metode LSB ini yang ingin dicapai adalah sebagai berikut:

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

Adapun manfaat dalam penelitian ini yang diperoleh dari penerapan metodeLSBadalah sebagai berikut:

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 Kemanan Data

Pada zaman teknologi informasi sekarang, data atau informasi merupakan suatu asset yang sangat berharga dan harus dilindungi.Hal ini juga diikuti oleh kemajuan teknologi komputer.Kemajuan teknologi komputer membantu semua aspek kehidupan manusia. Dengan adanya kemajuan dalam teknologi informasi, komunikasi dan komputer maka kemudian muncul masalah baru, yaitu masalah keamanan akan data dan informasi dan dalam hal ini akan membuka peluang bagi orang-orang yang tidak bertanggung jawab untuk menggunakannya sebagai tindak kejahatan. Dan tentunya akan merugikan pihak tertentu. Dalam keamanan data ada beberapa aspek yang berkaitan dengan persyaratan kemanan yaitu(**Pabokory, 2015**):

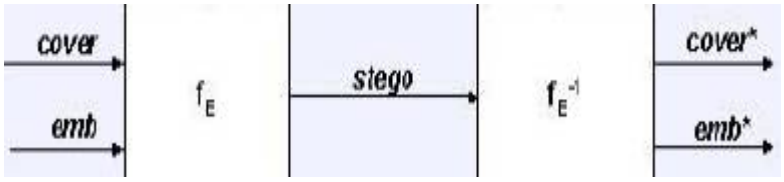
1. *Secrecy*. Berhubungan dengan akses membaca data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diakses dan dibaca oleh orang yang berhak.
2. *Integrity*. Berhubungan dengan akses merubah data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diubah oleh orang yang berhak.
3. *Availability*. Berhubungan dengan ketersediaan data dan informasi. Data dan informasi yang berada dalam suatu sistem komputer tersedia dan dapat dimanfaatkan oleh orang yang berhak. **(Pabokory, 2015).**

4. Lebih lanjut menurut **(Pabokory, 2015)**, terdapat lima langkah kemanan komputer yang baik untuk diperhitungkan yaitu; aset, analisis resiko, perlindungan, alat dan prioritas.

2.2 *Steganografi*

Steganografi adalah ilmu dan seni dari komunikasi yang tidak terlihat (Morkel, Eloff, & Olivier, 2005). 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.

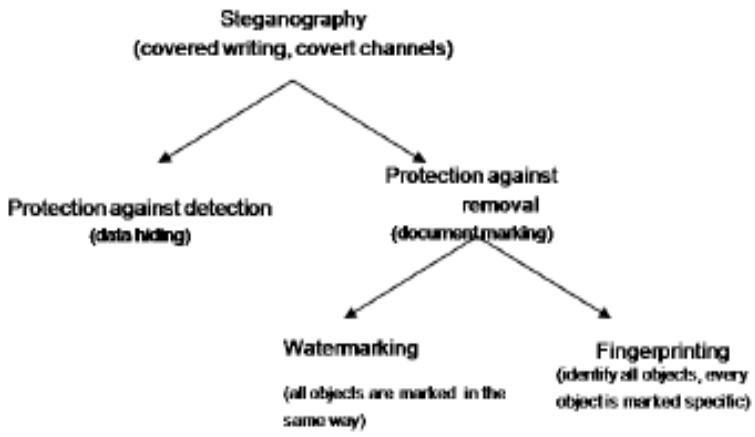


Gambar 2.1. Ilustrasi Sistem Steganografi.

(Sumber : Kumar & Pooja, 2013)

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.

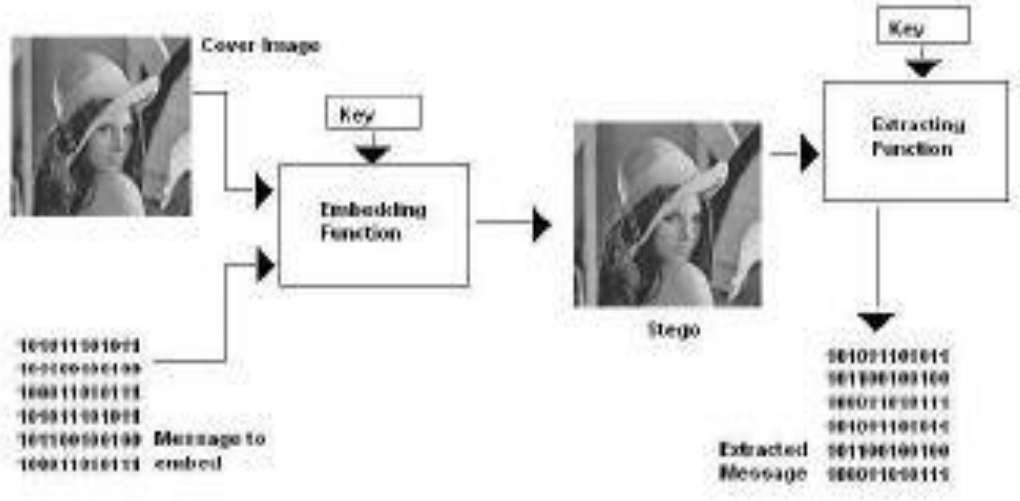


Gambar 2.2.Tipe dari Steganografi.

(Sumber : Kumar & Pooja, 2013)

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 2.3. Prosedur Steganografi.

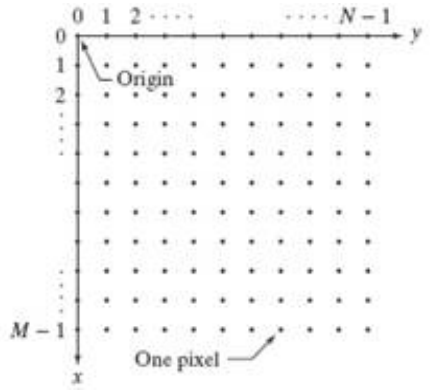
(*Sumber : Kumar & Pooja, 2013*)

2.3 *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. 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. (Taronisokhi, 2018)

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 2.4. Ilustrasi Citra Digital

(Sumber : Taronisokhi, 2018***)***

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

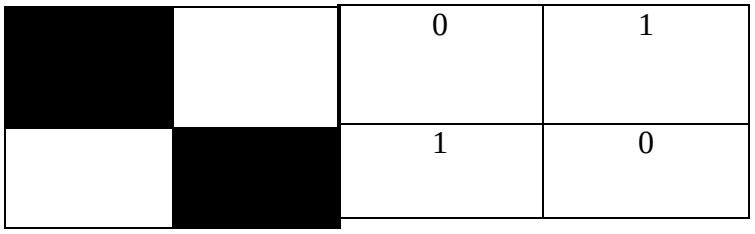
- x menyatakan posisi kolom;
- y menyatakan posisi baris;
- piksel pojok kiri-atas mempunyai koordinat (0, 0) dan piksel pada pojok kanan-bawah mempunyai koordinat (N-1, M-1).

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. Nilai 0 mewakili warna hitam, dan nilai 1 mewakili

warna putih, di mana setiap *pixel*-nya membutuhkan media penyimpanan sebesar 1 bit.



Gambar 2.5. **Contoh Citra Biner Berukuran 2x2 *Pixel***

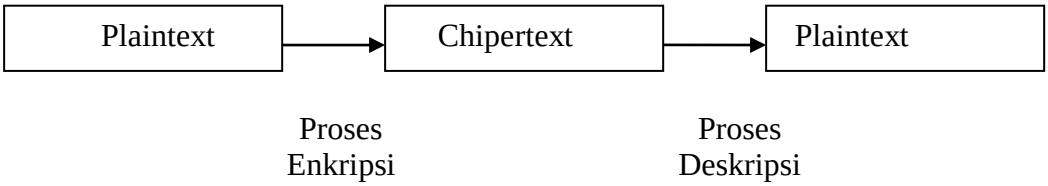
(Sumber : Darma Putra, 2010)

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. 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 bytemerepresentasikanwarna merah, hijau dan biru.

2.4 *Enkripsi*

Enkripsi merupakan hal yang sangat penting dalam *kriptografi* supaya keamanan data yang dikrimkan bisa terjaga kerahasiaannya.Pesan asli (plaintext) diubah menjadi kode-kode yang tidak dimengerti.*Enkripsi* bisa diartikan dengan chiper atau kode. Sama halnya dengan kita yang tidak mengerti sebuah kata, kita akan dapat melihatnya di dalam kamus atau daftar istilah-istilah. Berbeda halnya dengan *Enkripsi*, untuk mengubah plaintext ke bentuk chipertext, kita harus menggunakan algoritma yang dapat mengkodekan data yang kita inginkan. Berikut adalah penggambaran proses *Enkripsi*.



Gambar 2.6. Proses *Enkripsi*dan *Deskripsi*

(**Sumber: Pabokory, 2015)**

2.5 *Kriptografi* Klasik

Menurut (Bishop, 2014).[3]*kriptografi* klasik adalah *kriptografi* yang disebut juga sebagai *kriptografi* kunci tunggal atau *kriptografi* simetris yang menggunakan kunci yang sama untuk *Enkripsi* maupun *Deskripsi*. *Kriptografi* klasik merupakan *kriptografi* yang digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah ditemukan namun belum secanggih sekarang.*Kriptografi* ini melakukan pengacakan huruf pada kata terang / plaintext.

2.6 *One Time Pad (OTP)*

Algoritma *One Time Pad* (OTP) merupakan algoritma berjenis *Symetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara*stream Cipher* yang berasal dari hasi XOR antara *bitplaintext* dan *bitkey*. Pada metode ini *plain text* diubah kedalam kode ASCII dan kemudian dikenakan operasi XOR terhadap kunci yang sudah diubah ke dalam kode ASCII.(**Hamokwarong,2014**).

One-time pad adalah salah satu stream *Cipher* klasik yang secara matematis terbukti sempurna aman. *Cipher* teksnya tidak mungkin dapat dipecahkan. Keamanan algoritma *one-time pad* terletak pada penggunaan barisan bilangan acak sejati (*trully random*) sebagai kunci enkripsi, panjang kunci sama dengan panjang pesan dan tidak ada perulangan kunci sebagaimana pada pada *Vernam Cipher* atau *Vigenere Cipher*. (**Munir, 2014**)

Sayangnya *one-time pad* tidak dapat diimplementasikan secara praktis sebab pembangkitan bilangan acak sejati tidak dapat diulang kembali di sisi penerima pesan.Oleh karena itu kunci (*pad*) harus dikirim melalui saluran komunikasi yang kedua (misalnya melalui kurir), sayangnya saluran kedua itu umumnya lambat dan ongkosnya mahal.One-time pad masih dapat diterapkan namun kunci yang berupa barisan bilangan acak diganti dengan barisan bilangan semi-acak (*pseudo-random*) dengan syarat barisan kunci itu tidak boleh berulang. (**Munir, 2014**)

2.8 Algoritma

Penyelesaian permasalahan dengan menggunakan alat bantu system computer paling tidak akan melibatkan lima tahapan, yaitu:

1. Analisis masalah
2. Merancang algoritma
3. Membuat program computer
4. Menguji hasil program computer
5. Dokumentasi

Poin kedua menerangkan bahwa dalam perancangan sebuah system computer dibutuhkan adanya perancangan algoritma.Sehingga setelahnya dapat dilanjutkan ke tahap-tahap berikutnya hingga dokumentasi.

Algoritma adalah Sistim kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada softwarekomputer. Software terbangun atas susunan program (silahkan baca mengenai pengertian program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka,Algoritma berperan penting dalam penyusunan program atau syntax tersebut.

PengertianAlgoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, Algoritma sangat berperan penting dalam pembangunan

suatu software. Dalam dunia sehari-hari, mungkin tanpa kita sadari Algoritma telah masuk dalam kehidupan kita.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu.

Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari sebutan Algorizm (Abu Abdullah Muhammad Ibn Musa Al Khwarizmi, ahli matematika Uzbeki

a. Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika
b. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.
c. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
d. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah.

Pembuatan algoritma harus selalu dikaitkan dengan:

a. Kebenaran algoritma
b. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

Kriteria Algoritma yang baik:

a. Tepat, benar, sederhana, standar dan efektif
b. Logis, terstruktur dan sistematis
c. Semua operasi terdefinisi

d. Semua proses harus berakhir setelah sejumlah langkah dilakukan
e. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

2.9 *Unified Modeling Language* (UML)

1. Pengenalan UML

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang berisi sintak dalam memodelkan sistem secara visual (**Haviluddin, 2015**). Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *DataFlow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktur, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan *real time* (waktu nyata).

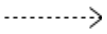
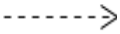
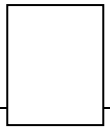
Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML).

2. *Use Case Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case*

digambarkan sebagai elips horizontal dalam suatu diagram *use case diagram* (**Haviluddin, 2015**).

Tabel 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (**Gellysa Urva, 2015**)

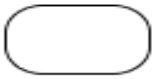
Contoh Use Case Diagram :

Gambar 2.7. Contoh *Use Case Diagram*
Sumber : (**Haviluddin, 2015)**

3. *Activity Diagram*

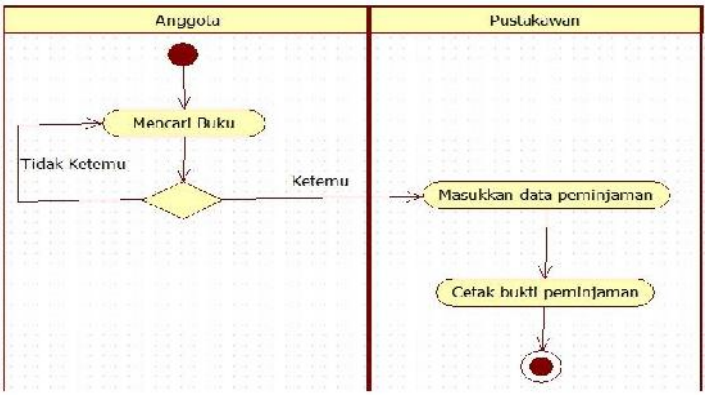
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2. Simbol *ActivityDiagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : (Gellysa Urva, 94 : 2015)

Contoh Activity Diagram :



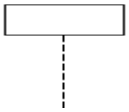
Gambar 2.8. Contoh *Activity Diagram*

Sumber : (Gellysa Urva, 94 : 2015)

4. *Sequence Diagram*

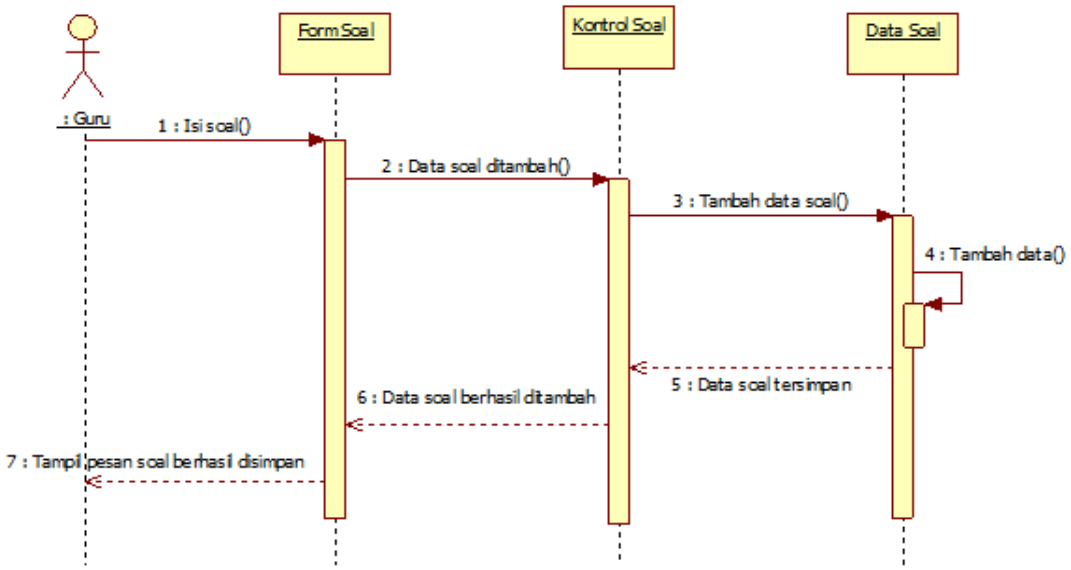
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

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 : (Gellysa Urva, 95 : 2015)

Contoh Squence Diagram :

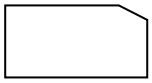


Gambar 2.9. Contoh *Squence Diagram*
Sumber : (Gellysa Urva, 95 : 2015)

5. *Class Diagram*

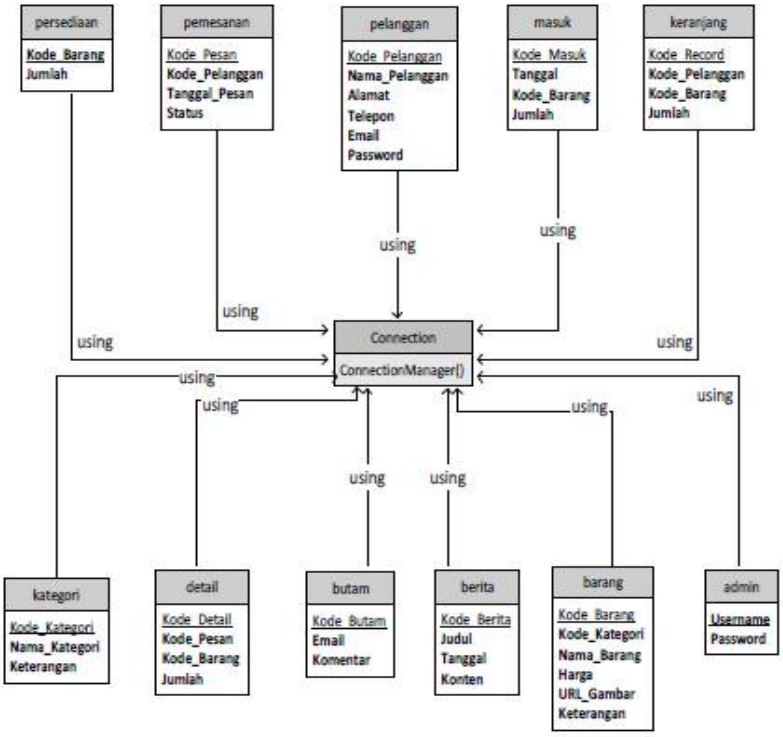
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Tabel 2.4. Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
2		*dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya
3		*extend*	Menspesifikasikan bahwa use case target memperluas prilaku dari use case sumber pada suatu titik yang diberikan.

Sumber : (Gellysa Urva, 95 : 2015)

Contoh *Class* Diagram :



Gambar 2.10. Contoh *Class Diagram*

2.10 Pengertian Informasi

Secara Etimologi, kata informasi ini berasal dari kata bahasa Perancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin yaitu *informationem* yang berarti "konsep, ide atau garis besar". Informasi ini merupakan kata benda dari *informare* yang berarti aktivitas dalam "pengetahuan yang dikomunikasikan".

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan berguna (**Yulansari, 2013**).

Informasi bisa menjadi fungsi penting dalam membantu mengurangi rasa cemas pada seseorang. Menurut pendapat (**Notoatmodjo, 2018)** bahwa semakin banyak memiliki informasi dapat memengaruhi atau menambah pengetahuan terhadap seseorang dan dengan pengetahuan tersebut bisa menimbulkan kesadaran yang akhirnya seseorang itu akan berperilaku sesuai dengan pengetahuan yang dimilikinya.

Informasi adalah data yang telah diolah melalui proses tertentu menjadi sesuatu yang menambah pengetahuan atau temuan yang mempunyai arti baru bagi pemakainya.

Adapun fungsi-fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.Informasi yang berkualitas harus akurat, tepat dan relevan.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

2.11 Pengertian Visual Studio

Visual Studio .Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. *Visual Studio .NET* menyediakan tools bagi para *developer* untuk membangun aplikasi yang berjalan di .*Net Framework* (**Safik, 2015**).

Visual Studio (*Beginners All-Purpose Symbolic Instruction Code*) merupakan Bahasa pemrograman *Integrated Development Environment* (IDE), yaitu bahasa pemrograman *visual* yang digunakan untuk membuat program aplikasi atau *software* berbasis sistem operasi *Microsoft Windows*, dengan menggunakan model pemrograman "*Common Object Model* (COM)".

Visual Studio merupakan turunan bahasa pemrograman *STUDIO* yang menawarkan pengembangan perangkat lunak komputer bebasis grafik dengan cepat. Dengan menggunakan bahasa pemrograman VB, para progammer dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan VB.

Microsoft Visual Studio (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (*COM*), *Visual Studio* merupakan turunan bahasa pemrograman *STUDIO* dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat, Beberapa bahasa skrip seperti *Visual Studio for Applications* (*VBA*) dan

Visual Studio Scripting Edition (*VBScript*), mirip seperti halnya *Visual Studio*, tetapi cara kerjanya yang berbeda.

Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Studio* Program-program yang ditulis dengan *Visual Studio* juga dapat menggunakan *Windows* API, tapi membutuhkan deklarasi fungsi luar tambahan.

Dalam pemrograman untuk bisnis, *Visual Studio* memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Studio*, yang diikuti oleh *C++, JavaScript, C#,* dan *Java*.

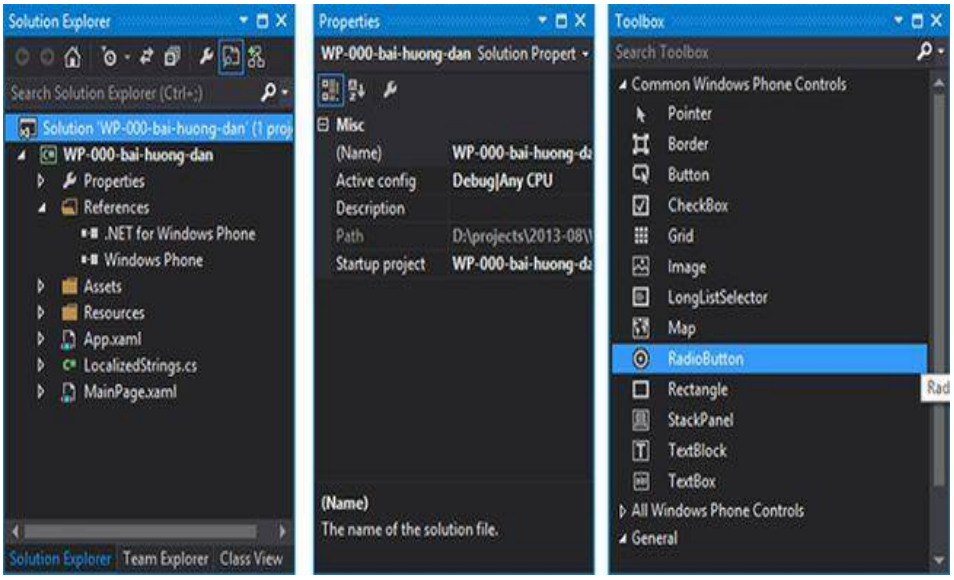
1. Komponen kerja

Beberapa komponen kerja program *visual Studio 2015* telah ditampilkan sebagai tampilan standard.Masih banyak lagi komponen yang masih tersembunyi sehingga memerlukan perintah tertentu untuk menampilkannya.Kita dapat mengatur komponen di dalam program visual Studio 2015 sesuai dengan yang kita butuhkan. Berikut ini adalah beberapa komponen kerja dari visual Studio 2015adalah :

a. Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain mulai dari tombol *label*, *pointer*, *button*, dan lain-lain.Berikut ini adalah gambaran *toolbox* pada *visual Studio 2015* :

Berikut ini adalah *table* yang berisi nama tombol yang terdapat didalam *toolbox* beserta fungsinya.



Gambar 2.11. Tampilan *Toolbox*
Sumber : (**Safik, 2015**).

Table 2.5.*Toolbox Visual Studio*

Nama tombol	fungsi
Pointer	Memilih, mengatur ukuran dan memindahkan posisi yang terpasang di bagian form.
Bindingsources	Untuk mengkoneksikan program ke database
Label	Menampilkan teks, dimana pengguna program tidak bisa mengubah teks tersebut
Groupbox	Untuk mengelompokkan item yang ada di form
Checkbox	Membuat kotak periksa, dimana pengguna program dapat memilih sekaligus
Listbox	Membuat daftar pilihan
Timer	Membuat control waktu dan interval yang diperlukan
Image	Menampilkan gambar pada form dalam format *bit*map, icone, atau metafile
Picturebox	Menampilkan gambar dari sebuah file
Textbox	Membuat teks, dimana teks tersebut dapat diubah oleh pembuat program
Button	Membuat tombol perintah
Combobox	Menambahkan control kotak combo yang merupakan control gabungan antara textbox dan listbox

Sumber : (**Safik, 2015**).

2.12 Tabel ASCII

ASCII merupakan kepanjangan dari (American Standard Code for Information Interchange), dan pengertian dari ASCII sendiri adalah suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-karakter angka maupun huruf didalam komputer, sebagai contoh dapat kita lihat pada karakter 1, 2, 3, A, B, C, dan sebagainya.

DEC	OCT	HEX	BIN	Symbol
0	000	00	00000000	NUL
1	001	01	00000001	SOH
2	002	02	00000010	STX
3	003	03	00000011	ETX
4	004	04	00000100	EOT
5	005	05	00000101	ENQ
6	006	06	00000110	ACK
7	007	07	00000111	BEL
8	010	08	00001000	BS
9	011	09	00001001	HT
10	012	0A	00001010	LF
11	013	0B	00001011	VT
12	014	0C	00001100	FF
13	015	0D	00001101	CR
14	016	0E	00001110	SO
15	017	0F	00001111	SI

16	020	10	00010000	DLE
17	021	11	00010001	DC1
18	022	12	00010010	DC2
19	023	13	00010011	DC3
20	024	14	00010100	DC4
21	025	15	00010101	NAK
22	026	16	00010110	SYN
23	027	17	00010111	ETB
24	030	18	00011000	CAN
25	031	19	00011001	EM
26	032	1A	00011010	SUB
27	033	1B	00011011	ESC
28	034	1C	00011100	FS
29	035	1D	00011101	GS
30	036	1E	00011110	RS
31	037	1F	00011111	US
DEC	OCT	HEX	BIN	Symbol
32	040	20	00100000	
33	041	21	00100001	!
34	042	22	00100010	"
35	043	23	00100011	#
36	044	24	00100100	$
37	045	25	00100101	%
38	046	26	00100110	&
39	047	27	00100111	'
40	050	28	00101000	(
41	051	29	00101001	)
42	052	2A	00101010	*
43	053	2B	00101011	+
44	054	2C	00101100	,
45	055	2D	00101101	-

46	056	2E	00101110	.
47	057	2F	00101111	/
48	060	30	00110000	0
49	061	31	00110001	1
50	062	32	00110010	2
51	063	33	00110011	3
52	064	34	00110100	4
53	065	35	00110101	5
54	066	36	00110110	6
55	067	37	00110111	7
56	070	38	00111000	8
57	071	39	00111001	9
58	072	3A	00111010	:
59	073	3B	00111011	;
60	074	3C	00111100	<
61	075	3D	00111101	=
62	076	3E	00111110	>
63	077	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C
68	104	44	01000100	D
69	105	45	01000101	E
70	106	46	01000110	F
71	107	47	01000111	G
72	110	48	01001000	H
73	111	49	01001001	I
74	112	4A	01001010	J
75	113	4B	01001011	K
76	114	4C	01001100	L

77	115	4D	01001101	M
78	116	4E	01001110	N
79	117	4F	01001111	O
80	120	50	01010000	P
81	121	51	01010001	Q
82	122	52	01010010	R
83	123	53	01010011	S
84	124	54	01010100	T
85	125	55	01010101	U
86	126	56	01010110	V
87	127	57	01010111	W
88	130	58	01011000	X
89	131	59	01011001	Y
90	132	5A	01011010	Z
91	133	5B	01011011	[
92	134	5C	01011100	\
93	135	5D	01011101	]
94	136	5E	01011110	^
95	137	5F	01011111	_
96	140	60	01100000	`
97	141	61	01100001	a
98	142	62	01100010	b
99	143	63	01100011	c
100	144	64	01100100	d
101	145	65	01100101	e
102	146	66	01100110	f
103	147	67	01100111	g
104	150	68	01101000	h
105	151	69	01101001	i
106	152	6A	01101010	j
107	153	6B	01101011	k

108	154	6C	01101100	l
109	155	6D	01101101	m
110	156	6E	01101110	n
111	157	6F	01101111	o
112	160	70	01110000	p
113	161	71	01110001	q
114	162	72	01110010	r
115	163	73	01110011	s
116	164	74	01110100	t
117	165	75	01110101	u
118	166	76	01110110	v
119	167	77	01110111	w
120	170	78	01111000	x
121	171	79	01111001	y
122	172	7A	01111010	z
123	173	7B	01111011	{
124	174	7C	01111100	|
125	175	7D	01111101	}
126	176	7E	01111110	~
127	177	7F	01111111	

128	200	80	10000000	€
129	201	81	10000001	
130	202	82	10000010	,
131	203	83	10000011	*f*
132	204	84	10000100	„
133	205	85	10000101	…
134	206	86	10000110	†
135	207	87	10000111	‡
136	210	88	10001000	ˆ
137	211	89	10001001	‰
138	212	8A	10001010	Š

139	213	8B	10001011	‹
140	214	8C	10001100	Œ
141	215	8D	10001101	
142	216	8E	10001110	Ž
143	217	8F	10001111	
144	220	90	10010000	
145	221	91	10010001	‘
146	222	92	10010010	’
147	223	93	10010011	“
148	224	94	10010100	”
149	225	95	10010101	•
150	226	96	10010110	–
151	227	97	10010111	—
152	230	98	10011000	˜
153	231	99	10011001	™
154	232	9A	10011010	š
155	233	9B	10011011	›
156	234	9C	10011100	œ
157	235	9D	10011101	
158	236	9E	10011110	ž
159	237	9F	10011111	Ÿ
160	240	A0	10100000	
161	241	A1	10100001	¡
162	242	A2	10100010	¢
163	243	A3	10100011	£
164	244	A4	10100100	¤
165	245	A5	10100101	¥
166	246	A6	10100110	¦
167	247	A7	10100111	§
168	250	A8	10101000	¨
169	251	A9	10101001	©

170	252	AA	10101010	ª
171	253	AB	10101011	«
172	254	AC	10101100	¬
173	255	AD	10101101	
174	256	AE	10101110	®
175	257	AF	10101111	¯
176	260	B0	10110000	°
177	261	B1	10110001	±
178	262	B2	10110010	²
179	263	B3	10110011	³
180	264	B4	10110100	´
181	265	B5	10110101	µ
182	266	B6	10110110	¶
183	267	B7	10110111	·
184	270	B8	10111000	¸
185	271	B9	10111001	¹
186	272	BA	10111010	º
187	273	BB	10111011	»
188	274	BC	10111100	¼
189	275	BD	10111101	½
190	276	BE	10111110	¾
191	277	BF	10111111	¿
192	300	C0	11000000	À
193	301	C1	11000001	Á
194	302	C2	11000010	Â
195	303	C3	11000011	Ã
196	304	C4	11000100	Ä
197	305	C5	11000101	Å
198	306	C6	11000110	Æ
199	307	C7	11000111	Ç
200	310	C8	11001000	È

201	311	C9	11001001	É
202	312	CA	11001010	Ê
203	313	CB	11001011	Ë
204	314	CC	11001100	Ì
205	315	CD	11001101	Í
206	316	CE	11001110	Î
207	317	CF	11001111	Ï
208	320	D0	11010000	Đ
209	321	D1	11010001	Ñ
210	322	D2	11010010	Ò
211	323	D3	11010011	Ó
212	324	D4	11010100	Ô
213	325	D5	11010101	Õ
214	326	D6	11010110	Ö
215	327	D7	11010111	×
216	330	D8	11011000	Ø
217	331	D9	11011001	Ù
218	332	DA	11011010	Ú
219	333	DB	11011011	Û
220	334	DC	11011100	Ü
221	335	DD	11011101	Ý
222	336	DE	11011110	Þ
223	337	DF	11011111	ß
224	340	E0	11100000	à
225	341	E1	11100001	á
226	342	E2	11100010	â
227	343	E3	11100011	ã
228	344	E4	11100100	ä
229	345	E5	11100101	å
230	346	E6	11100110	æ
231	347	E7	11100111	ç

232	350	E8	11101000	è
233	351	E9	11101001	é
234	352	EA	11101010	ê
235	353	EB	11101011	ë
236	354	EC	11101100	ì
237	355	ED	11101101	í
238	356	EE	11101110	î
239	357	EF	11101111	ï
240	360	F0	11110000	ð
241	361	F1	11110001	ñ
242	362	F2	11110010	ò
243	363	F3	11110011	ó
244	364	F4	11110100	ô
245	365	F5	11110101	õ
246	366	F6	11110110	ö
247	367	F7	11110111	÷
248	370	F8	11111000	ø
249	371	F9	11111001	ù
250	372	FA	11111010	ú
251	373	FB	11111011	û
252	374	FC	11111100	ü
253	375	FD	11111101	ý
254	376	FE	11111110	þ
255	377	FF	11111111	ÿ

2.13 Pengolahan Citra Digital

a. 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, Pengolahan Citra Digital, 2010).

b. 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.

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

2.14 Metode LSB (Least Significant Bit)

Adapun algoritma dari metode LSB (Least Significant Bit) ini adalah sebagai berikut:a. Baca informasi file, tentukan dimana posisi akhir file.b. Tandai posisi ctrl-z(penanda) sebagai awal baris penyisipan pesan.c. Sisipkan pesan

dimulai dari posisi ctrl-z(penanda) hingga akhir pesan.d. Sisipkan ctrl-z(penanda) kedua pada akhir pesan. Teknik LSB tidak mengubah isi awal dari file yang disisipi. Sebagaicontoh, jika kita menyisipkan sebuah pesan kedalam sebuah dokumen, isi daridocument tidak akan berubah. Ini yang menjadi salah satu keunggulan metode LSB dibandingkan dengan metode steganografi yang lain. Karena disisipkanpada akhir file, pesan yang disisipkan tidak akan bersinggungan dengan isifile, hal ini menyebabkan integrasi data dari file yang disisipi tetap terjaga. Namun, metode LSB tidak dapat menyisipkan pesan berukuran sangatbesar karena dapat membuat citra berubah dan mencurigakan, baiknya pesantidak terlalu besar agar tidak mencurigakan.Sesuai dengan konsep algoritma Least Significant Bit (Lsb) pada steganografi makapenanda pesan diletakkan di awal sebelum pesan disisipkan.

Metode ini meyembunyikan pesan rahasia dengan cara manambahkan bit-bit pesan yang akan disembunyikan ke akhir file citra penampung. (Gunawan,2018)

Proses penyisipan pesan dengan metode Least Significant Bit dapat dituliskan dalam algoritma sebagai berikut: (Gunawan,2018)

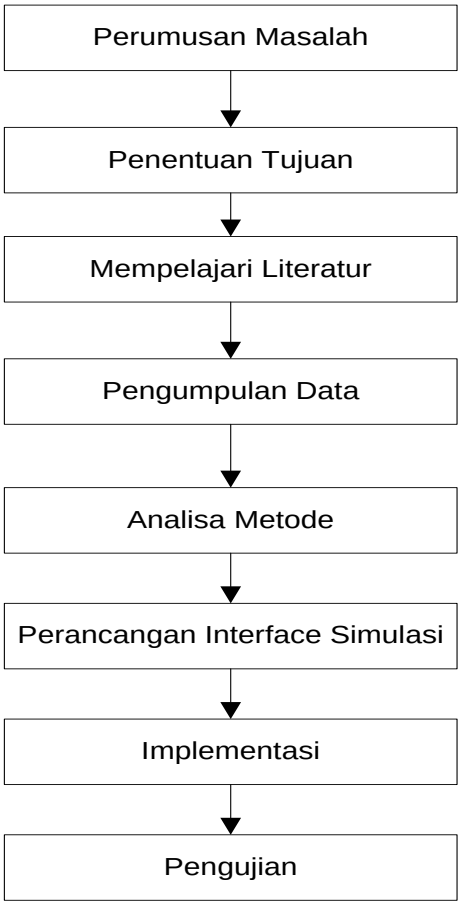
1. Inputkan pesan yang akan disisipkan
2. Ubah pesan menjadi kode-kode desimal.
3. Inputkan citra grayscale yang akan disisipi pesan.
4. Dapatkan nilai derajat keabuan masingmasing piksel.
5. Tambahkan kode desimal pesan sebagai nilai derajat keabuan citra.
6. Petakan menjadi citra baru.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Penerapan Algoritma Lsb (Least Significant Bit) Untuk Penyembunyian Teks Pada File Image 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*)

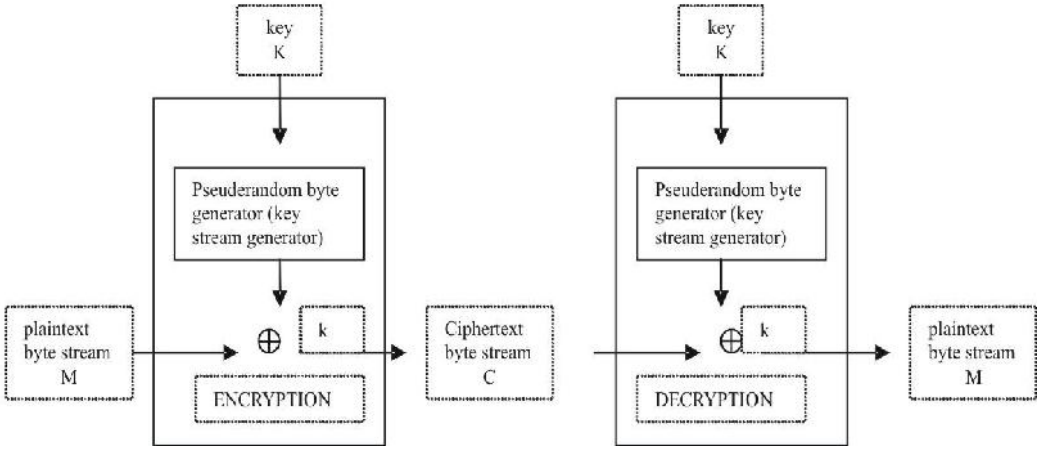
   Penulis melakukan pengamatan langsung pada setiap penggunan aplikasi chating yang sudah ada seperti WA, BBM dan Line untuk mengamati proses keamanan yang sudah dibuat sebelumnya.

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 Analisa 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 Sistem yang Berjalan

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

3.4 Analisa Proses Sistem Yang Dirancang

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.

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 teks nya ke dalam akhir biner Gambar penampung, sehingga akan terlihat seperti pada tabel berikut ini.

Tabel 3.3. Tabel biner Gambar yang berisi pesan rahasia

00000000	00000001	00010010	00000000	00000000	00010010	00000000	00000001	A
00010100	00000001	00000000	00010110	00000001	00000000	00011001	00000001	K
00000000	00011011	00000000	00000001	00011000	00000001	00000000	00011001	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	-

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.*

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.png" yang telah disisipkan *text* (*stego file*). Nilai setiap *pixel file* Gambar tersebut dapat dilihat pada Tabel 7.

Gambar 3.4. 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

00000000	00000001	00010010	00000000	00000000	00010010	00000000	00000001	A
00010100	00000001	00000000	00010110	00000001	00000000	00011001	00000001	K
00000000	00011011	00000000	00000001	00011000	00000001	00000000	00011001	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

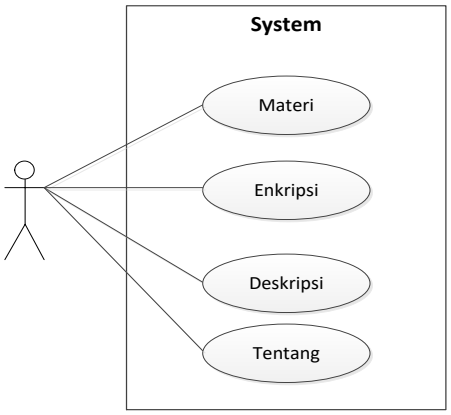
00000000	00000001	00010010	00000000	00000000	00010010	00000000	00000001	A
00010100	00000001	00000000	00010110	00000001	00000000	00011001	00000001	K
00000000	00011011	00000000	00000001	00011000	00000001	00000000	00011001	U

3.7 Perancangan Sistem

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.

3.7.1 Use case Diagram

Berikut adalah use case diagram yang menggambarkan kegiatan.



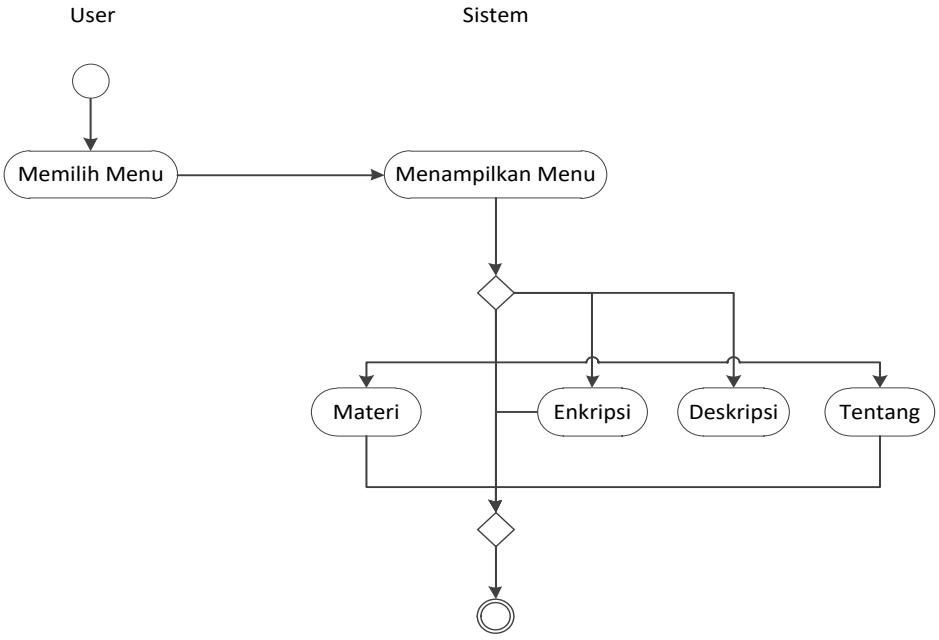
Gambar 3.5. Use Case Diagram

Keterangan :

Dalam use case diagram di atas, user/pengguna sebagai actor yang mempunyai use case Materi, Enkripsi dan Tentang. Halaman materi menamplikan data tentang materi dari LSB. Halaman enkripsi menampilkan proses enkripsi dari gambar menggunakan CRT dan Halaman deskripsi adalah kebalikan dari enkripsi.

3.7.2 Activity Diagram

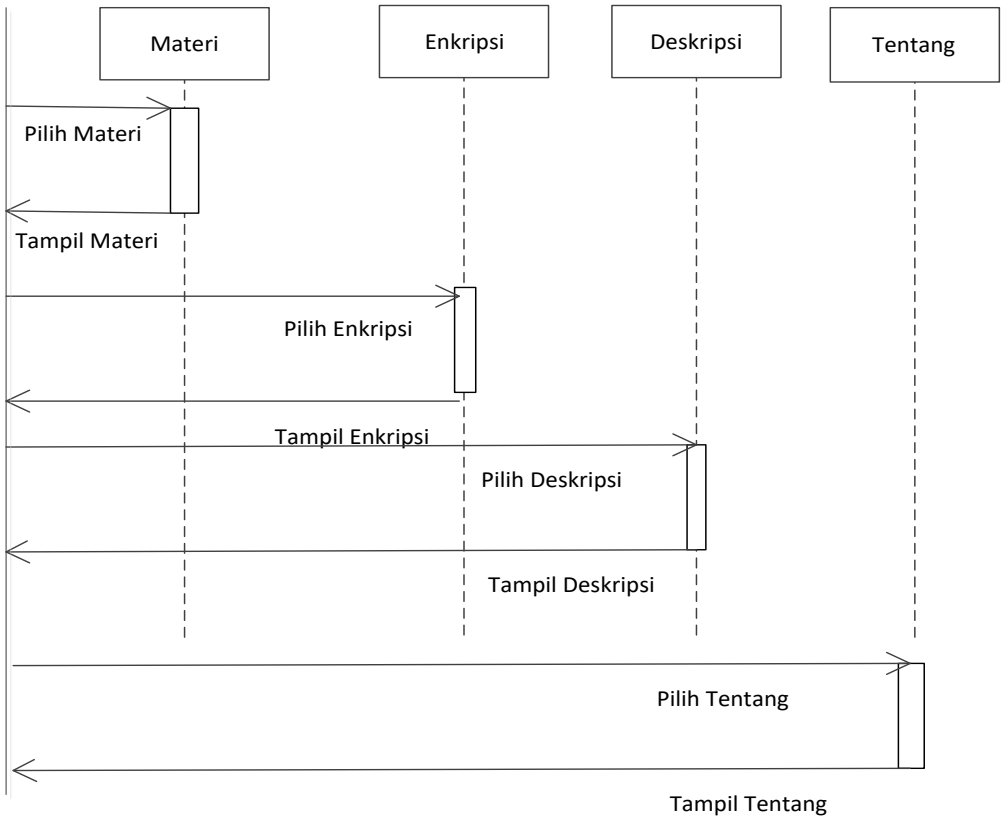
Activity diagram menggambarkan aktifitas-aktifitas yang terjadi dalam aplikasi dari aktivitas dimulai sampai aktivitas berhenti.



Gambar 3.6 Activity Diagram

User menjalankan program, lalu memilih menu yang dinginkan, sistem memberikan empat (4) menu, yaitu materi, enkripsi, deskripsi dan tentang. Materi memberikan info tentang materi teori tentang lsb. Enkripsi memberikan proses enkripsi file yang dilakukan sehinggan menghasilkan file enkripsi. Deskripsi memberikan fungsi mengembalikan file enkripsi menjadi file yang sebenarnya. Tentang memberikan info tentang pembuat program dan tujuan dari program ini.

3.7.3 Sequence Diagram



Gambar 3.7. *Sequence Diagram*

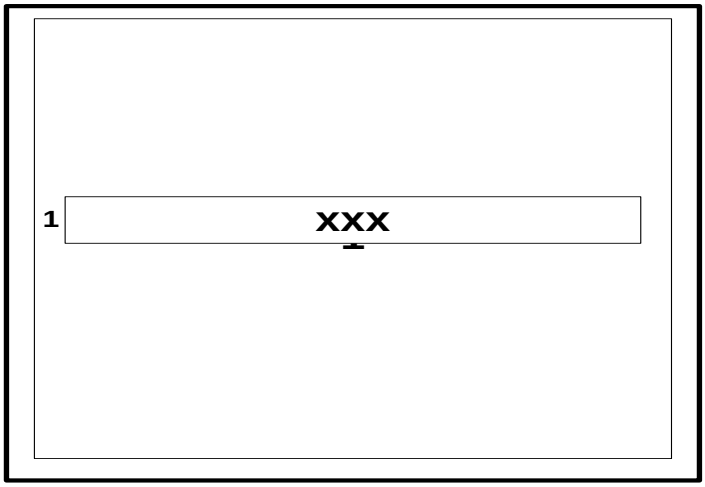
Keterangan Gambar :

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.

3.8 Perancangan Antarmuka

3.8.1 Rancangan Halaman Judul

Halaman judul merupakan halaman yang pertama muncul pada saat program dijalankan



Gambar 3.8 Rancangan Halaman Judul

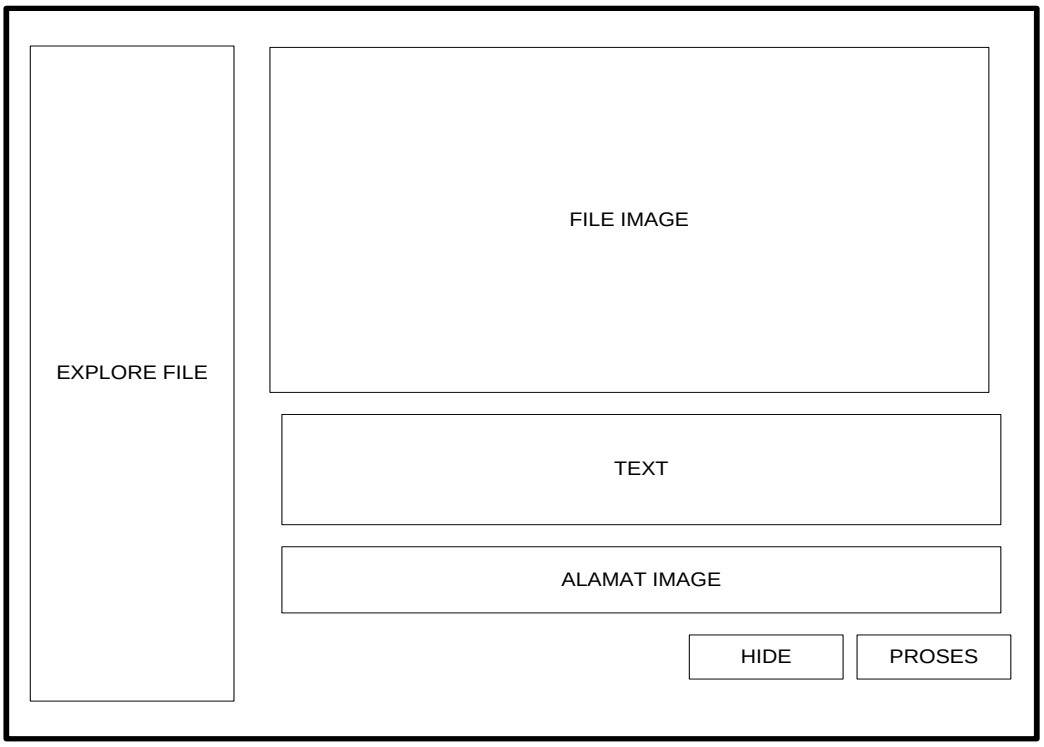
Pada rancangan di atas akan menampilkan judul yang kemudian akan pindah ke form menu utama dengan menggunakan timer.

Keterangan:

1. Berfungsi untuk menampilkan judul program.

3.8.4 Rancangan Halaman Penyembunyian

Berisi penjelasan mengenai Penyembunyian Text. Pengguna memasukkan tulisan asli atau *plaintext* Setelah itu, ditekan tombol Proses Hide yang kemudian akan menyimpan pesan tersebut.



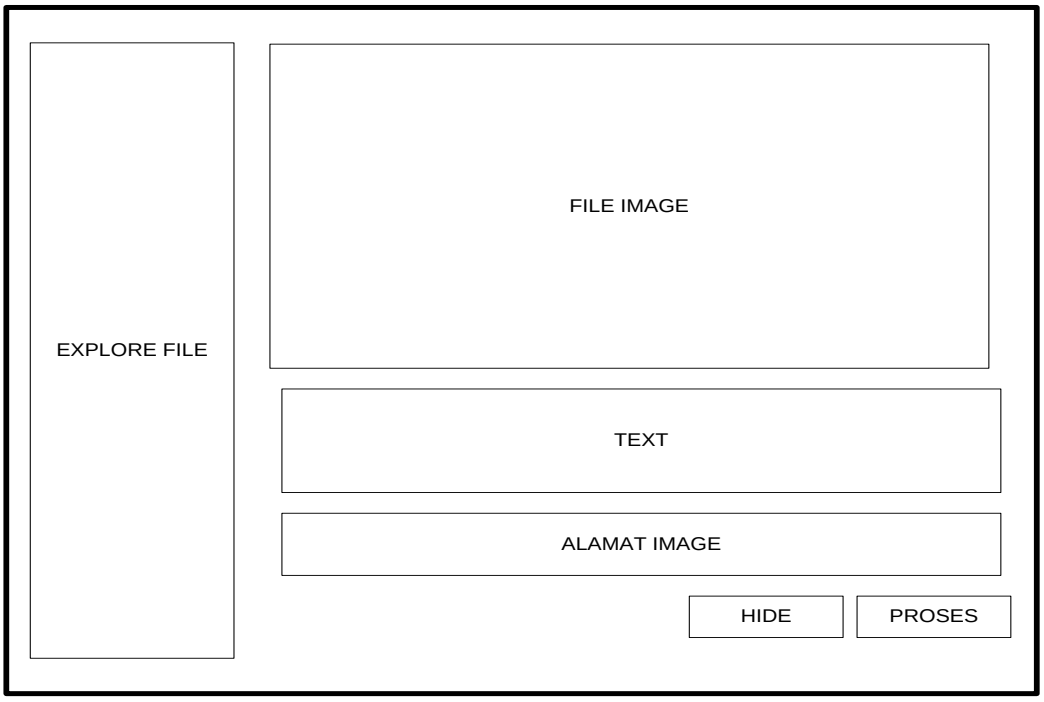
Gambar 3.11 Rancangan Halaman Penyembunyian

Keterangan:

1. File Image berfungsi untuk melihat foto yang telah kita upload dari komputer.
2. Explore File berfungsi untuk menampilkan nama Gambar .
3. Berfungsi untuk melihat alamat image .
4. Berfungsi untuk menyembunyikan pesan.

3.8.5 Rancangan Halaman Ekstrak

Pada Halaman Ini berisi penjelasan mengenai Ekstrak. Setelah itu, ditekan tombol Proses yang kemudian akan menampilkan ciphertext atau tulisan yang telah di sembunyikan.



Gambar 3.12 Rancangan Halaman Ekstrak

Keterangan:

1. Berfungsi untuk menampilkan nama Gambar yang sudah di upload.
2. Berfungsi untuk menampilkan teks yang telah di sembunyikan.
3. Tombol yang berfungsi untuk mencari Gambar yang ingin di tampilkan.

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.

4.1.1 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 ← LSB Image (gambar yang telah berisi pesan)

2. 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 = 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
```

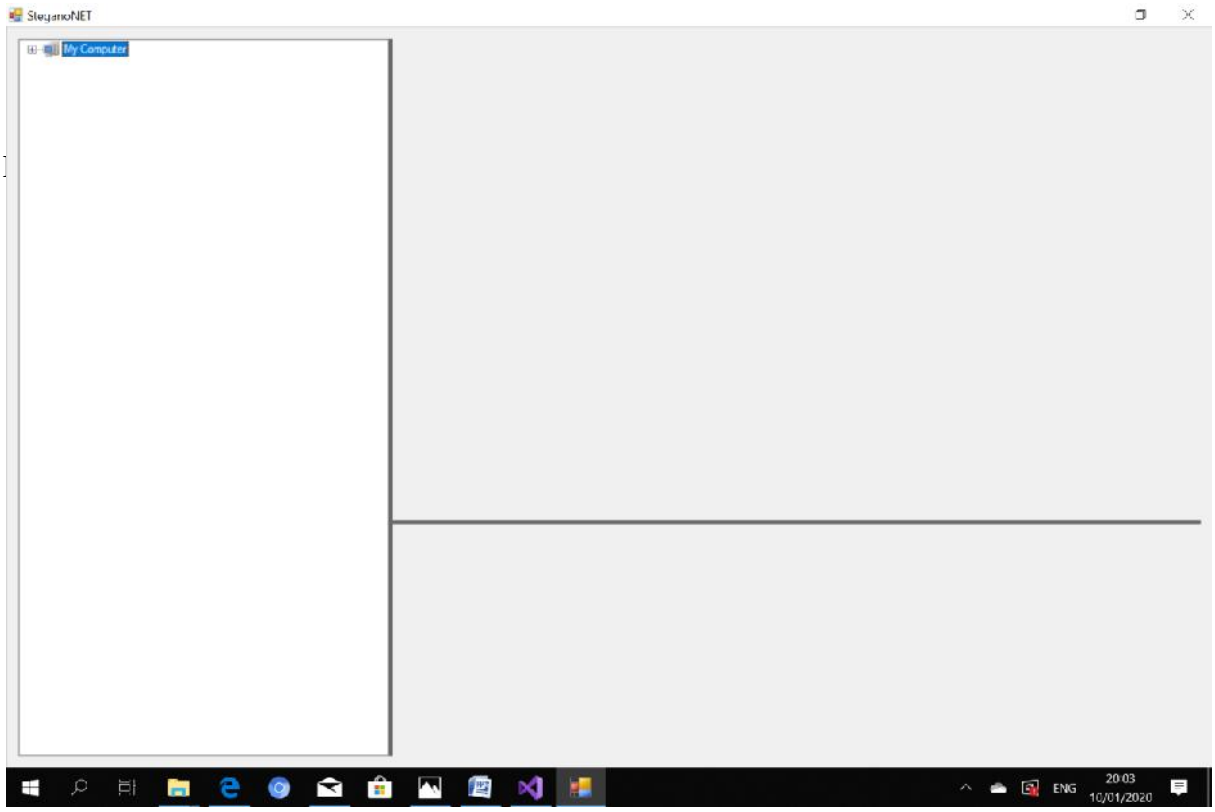
4.2 Implementasi Sistem

Tahap implementasi merupakan lanjutan dari tahap perancangan sistem. Pada tahap ini dilakukan implementasi sistem ke dalam 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.

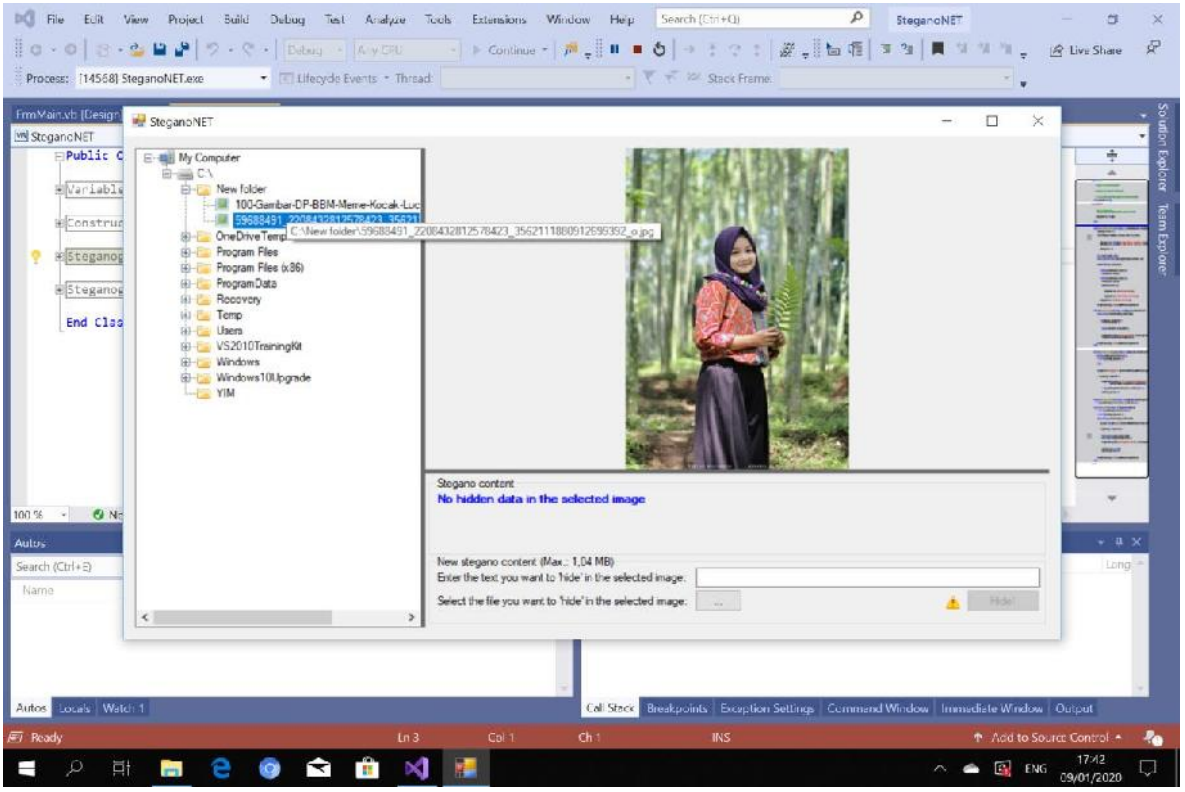
4.2.1 Tampilan Halaman Steganografi

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



4.2.2 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.



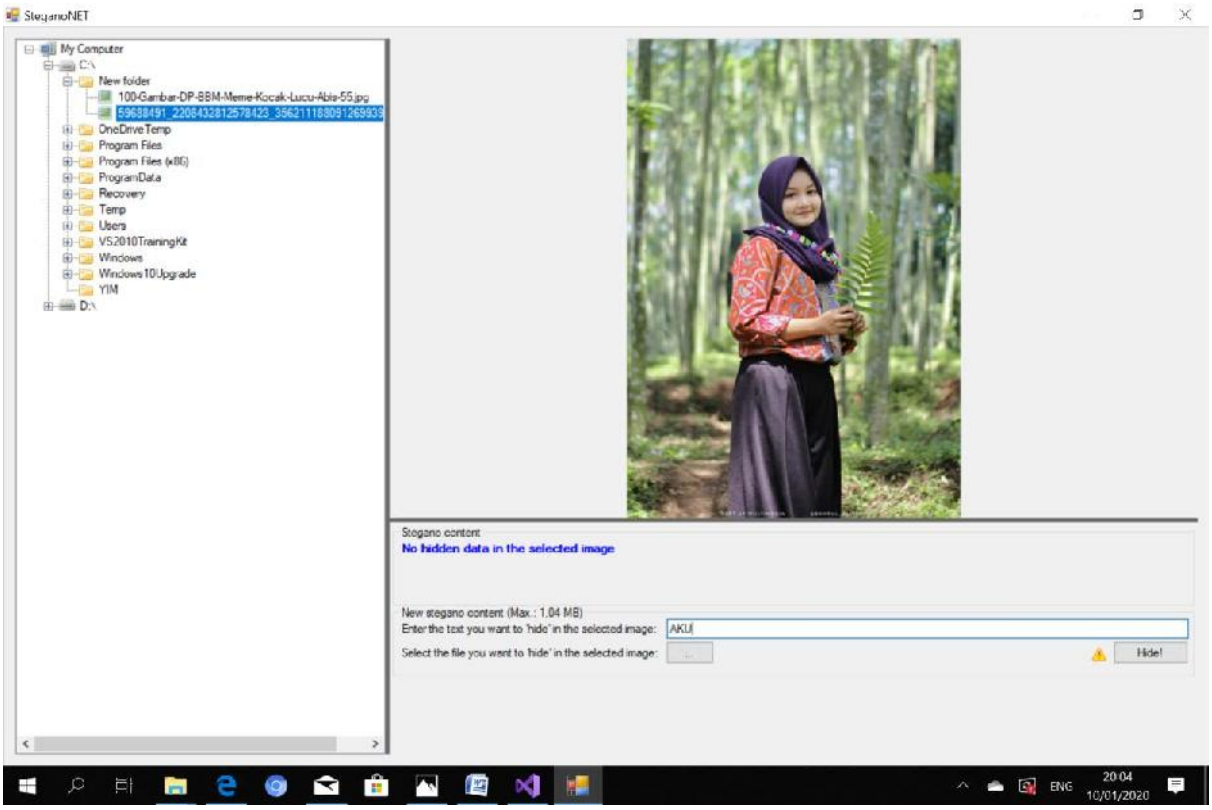
Gambar 4.2. Tampilan Cari Gambar

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.

4.2.3 Tampilan Penyembunyian Pesan Text

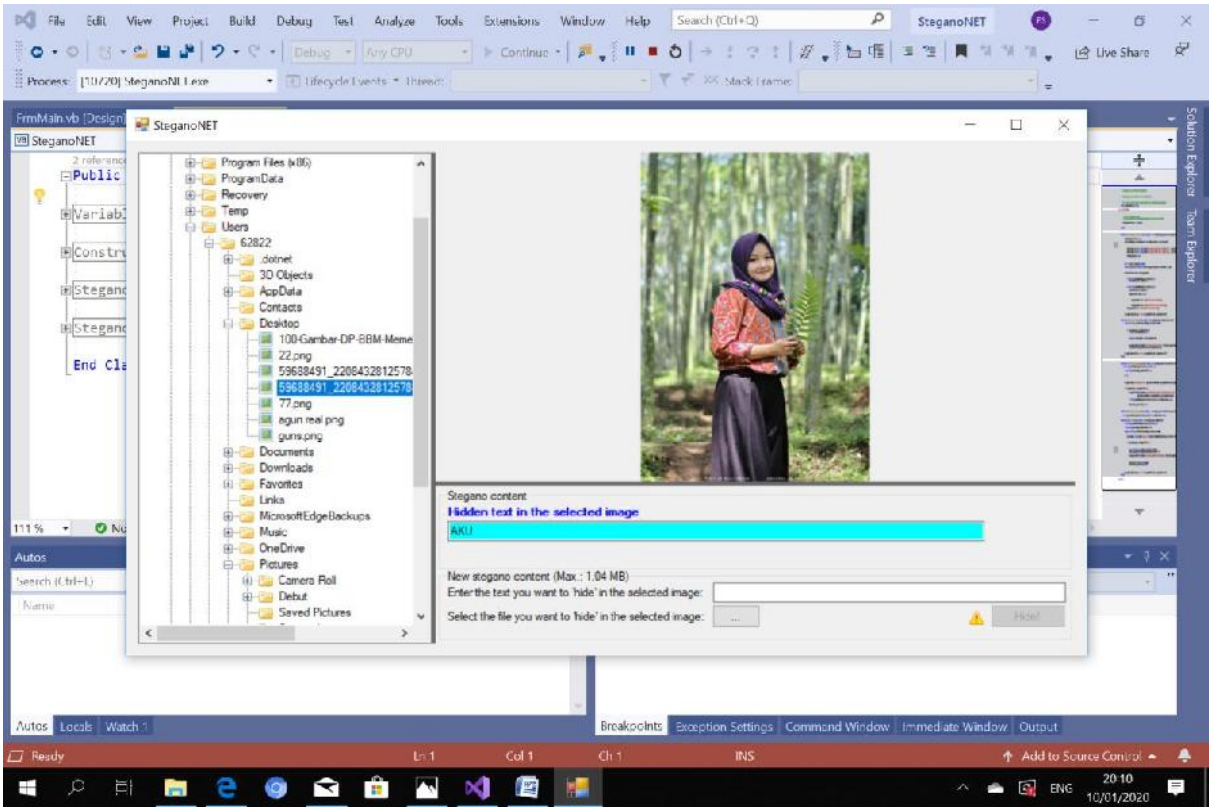
Halaman Penyembunyian Pesan Text merupakan halaman yang muncul pada saat proses untuk penyembunyian pesan. Tampilan halaman Penyembunyian Pesan Text dapat dilihat pada Gambar 17.



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 muncil pada *Picture Box* (Gambar LSB).



Gambar 4.4 Tampilan Gambar Yang Tersimpan Text

Keterangan:

1. Klik Buka pada *Button* "Buka Gambar LSB"
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.

4.3 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 sistem ini adalah metode pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak.

4.3.1 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 sistem 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**	**Jenis uji**
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	**Detai 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*

4.3.2 Rencana Pengujian

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 tabel 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 tabel 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 textbox tersebut, apakah textbox berfungsi dengan baik. Hasil uji dapat dilihat pada tabel 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 textbox tersebut, apakah textbox berfungsi dengan baik. Hasil uji dapat dilihat pada tabel 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 LSB.
Skenario	1. Aktor mengklik button 'Ambil Gambar LSB', 2. Lalu, masukkan password pada textbox password. 3. Setelah itu, klik button 'Ambil Pesan', jika sesuai password dengan gambar, maka pesan akan muncul pada textbox pesan.
Hasil yang didapat	Pesan text pada text box LSB.
Kesimpulan	Fungsi berjalan dengan baik

4.3.3 Kesimpulan dan hasil pengujian alpha

Hasil pengujian dari pengujian sistem telah selesai, menunjukkan bahwa sistem 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
Menampilkanl Pesan	Fungsi berjalan dengan baik
Input Gambar	Fungsi berjalan dengan baik
Input Pesan	Fungsi berjalan dengan baik
Input Password	Fungsi berjalan dengan baik

4.4 Kelebihan dan Kekurangan Sistem

Adapun kelebihan dan kekurangan dari system ini adalah sebagai berikut:

1. Kelebihan Sistem
   - Memberikan keamanan yang lebih baik.
   - Proses penginputan mudah dan friendly.
   - Proses keamanan menggunakan lsb yang mempersulit untuk di retas dan dirusak.
2. Kekurangan Sistem
   - Hanya melakukan embedded dan extraks
   - Sebaiknya dapat digunakan pada Android.
   - Hanya membahas teori LSB.

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 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

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.

Ariyus, Dony. 2006. *Computer Security*. Yogyakarta: Penerbit Andi.

Arjana, Putu H. dkk. 2012. *Implementasi Enkripsi Data Dengan Algoritma LSB*. Yogyakarta: Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012).

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.

Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." IT Journal Research and Development 2.1 (2017): 1-11.

Batubara, Supina, Sri Wahyuni, and Eko Hariyanto. "Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam." Seminar Nasional Royal (SENAR). Vol. 1. No. 1. 2018.

Bishop, Matt. 2005. *Introduction to Computer Security*. Boston: Addison-Wesley. Christensen, Chris. 2006. *Steganografi and LSB*. http:/www.nku.edu/~christensen/section%2014%20steganografi.pdf. Diakses pada 5 November 2016.

Fachri, B. (2018). Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 3, 98-102.

Fachri, B. (2018, September). APLIKASI PERBAIKAN CITRA EFEK NOISE SALT & PAPPER MENGGUNAKAN METODE CONTRAHARMONIC MEAN FILTER. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 87-92).

Fachri, B., Windarto, A. P., & Parinduri, I. (2019). Penerapan Backpropagation dan Analisis Sensitivitas pada Prediksi Indikator Terpenting Perusahaan Listrik. JEPIN (Jurnal Edukasi dan Penelitian Informatika), 5(2), 202-208.

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.

INDRA PERMANA, A. M. I. N. U. D. D. I. N. "SISTEM PAKAR MENDETEKSI HAMA DAN PENYAKIT TANAMAN KELAPA SAWIT PADA PT. MOEIS KEBUN SIPARE-PARE KABUPATEN BATUBARA." (2013).

Leong, Marlon. 2006. *Dari Programmer Untuk Proggrammer Visual Basic*.Yogyakarta: Penerbit Andi.

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.

Martin, Keith. 2012. *Everyday Cryptography*. Oxford: Oxford University Press.

Mulyana, Teady. 2012. *Steganografi Citra Digital Menggunakan Spreadsheet*. Vol: 8 No 2 Agustus 2012.

Pabokory, Fresly Nandar dkk. 2015. *Implementasi LSB Pengamanan Data Pada Pesan Teks, Isi File Gambar Menggunakan Algoritma Advanced Encryption Standard.* Vol: 10 No 1 Februari 2015.

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, Randi Rian. "IMPLEMENTASI METODE BACKPROPAGATION JARINGAN SARAF TIRUAN DALAM MEMPREDIKSI POLA PENGUNJUNG TERHADAP TRANSAKSI." JurTI (Jurnal Teknologi Informasi) 3.1 (2019): 16-20.

Rhee, Man Young. 1994. *Library of Congress Cataloging-in-Publication Data*. Singapore: McGraw-Hill Book Co.

Sutanto, Edhy. 2004. *Algoritma:Teknik Penyelesaian Permasalahan Untuk Komputasi*. Yogyakarta : Graha Ilmu.

Wahana Komputer. 2003. *Memahami Model Enkripsi dan Security Data*. Yogyakarta: Penerbit Andi.

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