IMPLEMENTASI KRIPTOGRAFI DALAM PENYISIPAN PESAN PADA CITRA DIGITAL MENGGUNAKAN METODE *PLAYFAIR CIPHER* DAN *LEAST SIGNIFICANT BIT* (LSB)

SKRIPSI

CHYNDY ASTIKA DANI HASIBUAN
0701162039

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021

IMPLEMENTASI KRIPTOGRAFI DALAM PENYISIPAN PESAN PADA CITRA DIGITAL MENGGUNAKAN METODE *PLAYFAIR CIPHER* DAN *LEAST SIGNIFICANT BIT* (LSB)

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

CHYNDY ASTIKA DANI HASIBUAN

0701162039

PROGRAM STUDI ILMU KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUMATERA UTARA

MEDAN

2021

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -

Kepada Yth.,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama	: Chyndy Astika Dani Hasibuan
Nomor Induk Mahasiswa	: 0701162039
Program Studi	: Ilmu Komputer
Judul	: Implementasi Kriptografi Dalam Penyisipan Pesan Pada Citra Digital Menggunakan Metode Playfair Cipher dan Least Significant Bit (LSB).

dapat disetujui untuk segera di*munaqasyah*kan. Atas perhatiannya kami ucapkan terimakasih.

Medan, 26 Maret 2021 M
12 Sya'ban 1442 H

Komisi Pembimbing,

Pembimbing Skripsi I,	Pembimbing Skripsi II,
Dr. Mhd. Furqan, S.Si, M.Comp.Sc	Yusuf Ramadhan Nasution, M.Kom
NIP. 198008062006041003	NIP. 1100000075

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama	: Chyndy Astika Dani Hasibuan
Nomor Induk Mahasiswa	: 0701162039
Program Studi	: Ilmu Komputer
Judul	: Implementasi Kriptografi Pada Penyisipan Pesan Pada Citra Digital Menggunakan Metode Playfair Cipher dan Least Significant Bit (LSB)

Dengan ini menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila dikemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 19 Maret 2021

Chyndy Astika Dani Hasibuan

NIM. 0701162039

KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: http://saintek.uinsu.ac.id, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor: B.097/ST/ST.V2/PP.01.1/05/2021

Judul	: Implementasi Kriptografi Dalam Penyisipan Pesan Pada Citra Digital Menggunakan Metode Playfair Cipher dan Least Significant Bit (LSB)
Nama	: Chyndy Astika Dani Hasibuan
Nomor Induk Mahasiswa	: 0701162039
Program Studi	: Ilmu Komputer
Fakultas	: Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal	: Jum'at, 26 Maret 2021
Tempat	: Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,
Ketua,

Ilka Zufria, M. Kom
NIP. 198506042015031006

Dewan Penguji,

Penguji I,

Dr. Mhd. Furqan, S.Si., M.Comp.Sc.
NIP. 198008062006041003

Penguji II,

Yusuf Ramadhan Nasution, M.Kom
NIP. 1100000075

Penguji III,

Rakhmat Kurniawan R, S.T, M.Kom
NIP. 198503162015031003

Penguji IV,

Abdul Halim Hasugian, M. Kom
NIP. 1100000113

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd. Syahnan, MA
NIP. 196609051991031002

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pengamanan pesan berupa teks menggunakan metode kriptografi Playfair cipher dengan Least Significant Bit (LSB) kedalam aplikasi berbasis website, mengetahui proses enkripsi dan deskripsi dengan Playfair cipher dan Least Significant Bit (LSB) terhadap pesan rahasia, dan mengetahui proses penyisipan pesan kedalam citra digital menggunakan metode Playfair Cipher dan Least Significant Bit (LSB) terhadap pesan rahasia. Metode penelitian yang digunakan adalah dengan metode analisis dan metode perancangan dimana pada metode analisis, penulis menganalisis kebutuhan dan melalui hasil analisis tersebut penulis merancang tahapan-tahapan yang akan dilakukan. Hasil yang dicapai adalah aplikasi dapat digunakan untuk menyembunyikan pesan rahasia pada citra digital dan perbedaan warna citra input dan citra hasil juga tidak kelihatan jelas. Simpulan yang didapat adalah setelah Playfair Cipher dan Least Significant Bit (LSB) diimplementasikan pada aplikasi, pesan rahasia dapat disembunyikan pada citra digital.

Kata Kunci: Kriptografi, Playfair Cipher, Least Significant Bit, Pesan Rahasia, Citra Digital

ABSTRACT

This research was aimed to design and implement a message security system in the form of text using the Playfair Cipher cryptographic method with the Least Significant Bit (LSB) into a website-based application, understand the encryption and description processes with the Playfair Cipher and Least Significant Bit (LSB) for secret messages, and understand the process of inserting messages into digital images using the Playfair Cipher and Least Significant Bit (LSB) to secret messages. The research method used is analysis and design method wherein the method of analysis, the authors analyze the needs and through the results of the analysis the authors design the stages to be carried out. The results of research are the application can be used to hide secret messages in digital images and the color difference between the input and output image is also not clear. After the Playfair Cipher and Least Significant Bit (LSB) are implemented in the application, the secret messages can be hidden in digital images.

Keywords: Cryptography, Playfair Cipher, Least Significant Bit, Secret Message, Digital Image

KATA PENGANTAR

Assalamu'alaikum wr. wb.

Syukur Alhamdulillah saya ucapkan kepada Allah SWT atas segala limpahan anugerah dan rahmat-Nya sehingga penulis dapat diselesaikan skripsi sebagaimana yang diharapkan. Tidak lupa shalawat berangkaikan salam kepada nabi Muhammad SAW yang telah memberi petunjuk bagi kehidupan manusia menuju jalan yang diridhoi Allah SWT. Skripsi yang berjudul "Implementasi Kriptografi Dalam Penyisipan Pesan Pada Citra Digital Menggunkan Metode Playfair Cipher dan Least Significant Bit (LSB) " dan akan diajukan untuk memenuhi salah satu persyaratan untuk memperoleh gelar sarjana S1 Fakultas Sains dan Teknologi Jurusan Ilmu Komputer UIN Sumatera Utara.

Penulis menyadari bahwa skripsi ini dapat diselesaikan berkat dukungan dan bantuan dari berbagai pihak. Oleh karena itu penulis berterima kasih kepada semua pihak yang secara langsung maupun tidak langsung dalam memberikan kontribusi untuk menyelesaikan skripsi ini. Penulis menyampaikan ucapan terima kasih kepada:

1. Bapak **Prof. Dr. H Syahrin Harahap, M.A** selaku rektor UIN Sumatera Utara yang telah memberikan fasilitas yang baik.
2. Bapak **Dr. Mhd Syahnan, M.A** selaku dekan Fakultas Sains dan Teknologi UIN Sumatera Utara yang telah memberikan kesempatan untuk peneliti menimba ilmu di jurusan Ilmu Komputer.
3. Bapak **Ilka Zufria, M.Kom.** selaku ketua jurusan Ilmu Komputer yang telah menyetujui judul ini.
4. Bapak **Rakhmat Kurniawan R, S.T.,M.Kom.** selaku sekretaris jurusan Ilmu Komputer yang telah menyetujui judul ini.
5. Ibu **Sriani, M.Kom.** selaku pembimbing akademik yang telah memberikan bimbingan dan arahan sehingga peneliti dapat menjalani studi akademik di UIN Sumatera Utara dengan baik.
6. Bapak **Dr. Mhd. Furqan, S.Si., M.Comp.Sc.** selaku dosen pembimbing I yang telah banyak memberikan arahan serta bimbingan kepada peneliti.
7. Bapak **Yusuf Ramadhan Nasution, M.Kom.** selaku dosen pembimbing II yang telah banyak memberikan waktu nya untuk membimbing peneliti.

8. Teristimewa peneliti sampaikan terima kasih dengan setulus hati kepada kedua orang tua tercinta, ayah **Sulaiman Hasibuan** dan mama **Nuriani Br. Purba** yang sampai detik ini telah berjuang membesarkan dan mendidik peneliti, berkat kasih sayangnya dan pengorbanan yang tak terhingga sehingga peneliti dapat menyelesaikan studi ke bangku perkuliahan.
9. Seluruh pihak yang tidak dapat saya sebutkan satu persatu yang telah membantu peneliti.

Semoga Allah SWT membalas semua kebaikan yang telah diberikan Bapak/Ibu serta Saudara/i, kiranya kita semua tetap berada dalam lindungan-Nya demi penyelesaian skripsi ini. Semoga proposal ini bermanfaat dalam memperkaya khazanah ilmu pengetahuan kita. Aamiin..

Wassalamu'alaikum Wr. Wb.

Medan, Maret 2021

Penulis,

Chyndy Astika Dani Hasibuan

NIM. 0701162039

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Keamanan informasi saat ini merupakan salah satu komponen terpenting di era perkembangan teknologi yang semakin pesat. Bagi beberapa orang, sebuah informasi termasuk sesuatu yang berharga maka dari itu terkadang seseorang tidak ingin orang lain mengetahui informasi tersebut. Namun sering kali informasi disalahgunakan oleh pihak yang tidak bertanggung jawab untuk meraih keuntungan atau pun hanya untuk merusaknya. Allah SWT juga menganjurkan untuk menjaga kerahasian yang harus dijaga hal ini terdapat didalam surat an-Nisa' ayat 58 yang berbunyi:

إِنَّ ٱللَّهَ يَأۡمُرُكُمۡ أَن تُؤَدُّواْ ٱلۡأَمَٰنَٰتِ إِلَىٰٓ أَهۡلِهَا وَإِذَا حَكَمۡتُم بَيۡنَ ٱلنَّاسِ
أَن تَحۡكُمُواْ بِٱلۡعَدۡلِۚ إِنَّ ٱللَّهَ نِعِمَّا يَعِظُكُم بِهِۦٓۗ إِنَّ ٱللَّهَ كَانَ سَمِيعَۢا بَصِيرٗا ٥٨

"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat. (QS. An-Nisa: 58)

Salah satu cara untuk mengatasi masalah keamanan informasi adalah dengan diterapkannya teknik kriptografi. Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci deskripsi. Tujuan adanya kriptografi ialah agar sebuah data yang disampaikan hanya diketahui oleh orang yang dituju ataupun yang berhak untuk mengetahuinya sehingga tidak disalah gunakan oleh orang-orang yang tidak bertanggung jawab.

Kriptografi terdapat berbagai macam teknik yang dapat digunakan dalam upaya pengamanan data salah satu nya ialah *Playfair Cipher*. *Playfair cipher*

tergolong dalam kriptografi klasik, *Playfair* merupakan *digraphs cipher* ialah setiap proses enkripsi dilakukan pada dua huruf atau pasangan huruf. *Playfair Cipher* mengenkripsi pasangan huruf, bukan huruf tunggal seperti pada cipher klasik lainnya. Tujuannya untuk membuat analisis frekuensi menjadi sulit sebab frekuensi kemunculan huruf didalam cipherteks akan menjadi datar. Salah satu teknik dalam penyembunyian pesan. Steganografi adalah suatu ilmu seni dalam menyembunyikan informasi dengan memasukkan informasi atau pesan tersebut ke dalam media lain. Sehingga keberadaan informasi tersebut tidak diketahui orang lain. Media yang dapat menampung informasi tersebut adalah citra digital, video, audio, dan teks.

Didalam dunia teknologi tidak cukup apabila hanya menggunakan kriptografi saja, ada baiknya menggunakan teknik steganografi. Steganografi juga merupakan lain yang mengetahuinya. Terdapat dua jenis dalam kriptografi yaitu kriptografi klasik dan kriptografi modern. Kriptografi klasik merupakan metode untuk mengubah data asli (*Plainteks*) ke bentuk sandi (*Cipherteks*) dengan menggunakan kunci yang sama. Sedangkan Kriptografi Modern ialah Metode yang menggunakan dua buah kunci yakni kunci publik (*Public key*) yang dapat dipublikasikan dan kunci privat (*Private key*) ialah kunci yang harus dirahasiakan.

Salah satu metode yang dapat digunakan dalam steganografi adalah *Least Significant Bit* (LSB). LSB merupakan teknik penyembunyian pesan pada lokasi bit terendah dalam citra digital. Pesan rahasia akan dikonversikan ke dalam bentuk biner dan disembunyikan kedalam sebuah media penyembunyian berupa citra digital. Hasil dari penyembunyian pesan menggunakan metode LSB tidak memiliki perubahan sehingga sulit dibedakan oleh mata manusia.

Dari latar belakang masalah diatas, penulis bermaksud untuk mengimplementasikan salah satu jenis kriptografi yaitu algoritma *Playfair Cipher* untuk memenuhi kebutuhan keamanan pesan rahasia dan akan dikombinasikan dengan metode steganografi yaitu *Least Significant Bit* (LSB).

Kombinasi dari teknik kriptografi *Playfair Cipher* dan steganografi LSB diharapkan dapat lebih meningkatkan pengamanan pada informasi yang bersifat

rahasia. Sebuah pesan rahasia akan dienkripsi terlebih dahulu menggunakan *Playfair Cipher* kemudian hasil kriptografi tersebut akan disembunyikan didalam citra digital menggunakan metode steganografi LSB.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat dibuat rumusan masalah sebagai berikut :

1. Bagaimana merancang dan mengimplementasikan sistem pengamanan pesan rahasia berupa teks menggunakan metode kriptografi *Playfair cipher* dengan *Least Significant Bit* (LSB) kedalam aplikasi berbasis website ?
2. Bagaimana melakukan proses enkripsi dan deskripsi dengan *Playfair cipher* dan *Least Significant Bit* (LSB) terhadap pesan rahasia?
3. Bagaimana proses penyisipan pesan kedalam citra digital menggunakan metode *Playfair Cipher* dan *Least Significant Bit* (LSB) ?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Pesan rahasia yang akan di proses berupa teks.
2. Data yang yang dienkripsi adalah file text berformat (*.*txt*) .
3. Panjang teks yang dapat dienkripsi maksimal 225 karakter.
4. Citra yang digunakan sebagai wadah penyisipan pesan dalam format *.jpg.
5. Citra gambar yang digunakan merupakan citra RGB.
6. Diimplementasikan berbasis web menggunakan bahasa pemrograman PHP.
7. Proses enkripsi dan deskripsi menggunakan metode *Playfair Cipher*.
8. Proses encoding dan decoding menggunakan metode *Least Significant Bit (LSB).*
9. Akan terjadi perubahan ukuran citra pada proses penyisipan pesan.
10. Pada saat proses deskripsi pesan asli akan berubah menjadi huruf kapital.

11. Citra yang digunakan sebagai penampung maksimal berukuran 700 x 700 pixel

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penyusunan penelitian ini adalah sebagai berikut :

1. Merancang dan mengimplementasikan sistem pengamanan pesan berupa teks menggunakan metode kriptografi *Playfair cipher* dengan *Least Significant Bit* (LSB) kedalam aplikasi berbasis website .
2 Untuk mengetahui proses enkripsi dan deskripsi dengan *Playfair cipher* dan *Least Significant Bit* (LSB) terhadap pesan rahasia.
3 Untuk mengetahui proses penyisipan pesan kedalam citra digital menggunakan metode *Playfair Cipher* dan *Least Significant Bit* (LSB) terhadap pesan rahasia.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penyusunan penelitian ini adalah :

1. Dapat merancang dan mengimplementasikan sistem pengamanan pesan berupa teks menggunakan metode kriptografi *Playfair cipher* dengan *Least Significant Bit* (LSB) kedalam aplikasi berbasis website .
2. Dapat mengetahui dan melakukan enkripsi dan deskripsi dengan *Playfair cipher* dan *Least Significant Bit* (LSB) terhadap pesan rahasia.
3. Dapat mengetahui dan melakukan proses penyisipan pesan kedalam citra digital menggunakan metode *Playfair Cipher* dan *Least Significant Bit* (LSB) terhadap pesan rahasia.

1.6 Referensi Penelitian Sebelumnya

Adapun beberapa penelitian sebelumnya yang berkaitan dengan pengamanan pesan rahasia menggunakan kriptograafi *playfair cipher* dan steganografi *least significant bit* (LSB) adalah sebagai berikut :

1. Ratna Wati Simbolon (2016) didalam jurnalnya yang berjudul Pengamanan Transkrip Nilai Mahasiswa Menggunakan Kriptografi Playfair Cipher dan Steganografi dengan Teknik Least Significant Bit (LSB). Kriptografi playfair cipher mengenkripsi pasangan huruf dengan tujuanmembuat analisis frekuensi menjadi sulit sebab frekuensi kemunculan huruf di dalam ciphertext akan menjadi datar. Penerapan kode ASCII juga akan membuat hasil enkripsi semakin sulit untuk dimengerti oleh pihak ketiga. Kombinasi steganografi menggunakan LSB dengan hasil file citra bitmap grayscale 8 bit per piksel dengan format biner akan menghindari kecurigaan. Dengan adanya penerapan kombinasi kriptografi dan steganografi dalam pengamanan data transkrip nilai mahasiswa dipastikan tidak akan diketahui oleh orang lain karena tidak akan mengandung kecurigaan.
2. Michel Sitorus (2015) didalam jurnalnya yang berjudul Teknik Steganografi Dengan Metode Least Significant Bit (LSB). Berdasarkan penelitian dari teknik penyembunyian data dengan metode LSB, maka metode LSB adalah metode yang sederhana dan mudah di aplikasikan ke dalam sistem yang membutuhkan penyisipan data ke dalam gambar. Dengan menggunakan teknik penyembunyian data ke dalam gambar dapat menjadi media untuk pengamanan data yang akan di kirim. Data rahasia berupa teks dapat di sisipkan ke dalam gambar dengan kunci yang di buat dan di mengerti oleh pengguna aplikasi. Steganography dapat di implementasikan untuk proses otentikasi data dan di gunakan untuk komunikasi atau pertukaran data yang rahasia. Penggunaan steganography dapat bermanfaat dan mencegah kebocoran informasi dari proses penyadapan.
3. Sugeng Murdowo (2020) Manual Perhitungan Menggunakan Kriptografi Klasik Playfair Cipher. Dari pembahasan yang telah diuraiakan dimana Algoritma Playfair Chiper dengan menggunakan kata kunci yang panjang dan menggunakan bujur sangkar 6X6. Mampu menghasilkan enkripsi pada satu kalimat plaintext yang cukup panjang menjadi chipertext yang cukup

akurat dan cukup membingungkan bagi penerima pesan sebelum pesan tersebut dilakukan deskripsi

4. Wiyata (2016) didalam jurnalnya yang berjudul implementasi steganografi Metode LSB Menggunakan Program PHP untuk keamanan Pesan Gambar. Metode LSB dapat menyembunyikan pesan yang sulit untuk dipecahkan. Kemudian citra digital yang disisipkan dengan metode LSB ditambah dengan enkripsi akan semakin sulit untuk dipecahkan oleh orang yang tidak berkepentingan.
5. Desumeri Laoli, dkk (2020) Penerapan Hill Cipher dan Least Significant Bit (LSB) Untuk Pengamanan Pesan Pada Citra Digital. Berdasakan dari analisa, perancangan dan implementasi pada aplikasi pengamanan pesan pada citra digital dengan menggabungkan metode Least Significant Bit (LSB) dan algoritma hill cipher, menghasilkan Proses pengamanan pesan pada citra digital aman dan tidak diketahui secara kasat mata, karena besar dari bitmap hasil steganografi tidak terlihat secara signifikan perubahan setelah dari proses penyisipan biner teks ke dalam biner bitmap menggunakan metode least significant bit (LSB) yaitu penggantian bit terakhir sehingga kapasitas dari bitmap sebelum dan sesudah disteganografi tidak mengalami perubahan yang signifikan. Pengujian pesan teks menggunakan algoritma hill cipher berhasil dilakukan sesuai tepat dengan alur atau langkah-langkahnya sehingga menghasilkan cipherteks yang berupa pengacakan huruf abjad. Pesan yang akan diambil dari bitmap dapat dilanjutkan ke proses dekripsi yang bertujuan untuk mengembalikan pesan ciphet text ke bentuk semula (plain text) melalui proses dekripsi algoritma hill cipher yang sesuai.

BAB II
TINJAUAN PUSTAKA

2.1. Kriptografi

Kriptografi berasal dari bahasa Yunani "*cryptos*" artinya rahasia *(secret),* sedangkan *"graphein"* artinya tulisan rahasia *(writing).* Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman (Schneier, 1996). Ada beberapa definisi kriptografi yang telah dikemukakan di dalam berbagai literatur. Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti maknanya. Definisi ini mungkin cocok pada masa lalu dimana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih sekedar *privacy*, tetapi juga untuk tujuan data *integrity*, *autentication,* dan non-repudiation (Nurul Fitriani Andi Mu'mi).

Kriptografi merupakan salah satu teknik dari beberapa teknik keamanan data yang sering digunakan untuk mengamankan data, seperti halnya menyandikan pesan asli (*plaintext*) ke dalam bentuk pesan rahasia (*ciphertext*). Proses pengaman ini melibatkan algoritma dan kunci. Kunci yang telah dienkripsi dapat dengan mudah dideskripsi kembali, yaitu dari *ciphertext* menjadi *plaintext*. Oleh karena itu diperlukan algoritma kriptografi yang kuat. Namun teknik ini masih menimbulkan kecurigaan pada orang lain yang melihat pesan tersebut.

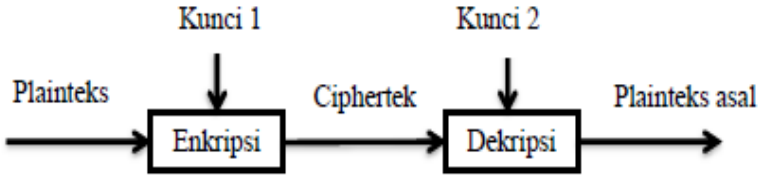
Menurut Munir (munir,2010) Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Keamanan pesan diperoleh dengan menyandikannya menjadi pesan yang tidak mempunyai makna. Pesan yang dirahasiakan dinamakan plainteks, sedangkan pesan hasil penyandian disebut cipherteks. Proses penyandian plainteks menjadi cipherteks disebut enkripsi dan proses membalikkan cipherteks menjadi plainteks asalnya disebut dekripsi.

Algoritma kriptografi adalah urutan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar yaitu :

1. Enkripsi: merupakan hal yang sangat penting dalam kriptografi pengamanan data yang dikirimkan agar terjaga kerahasiaanya. Pesan asli disebut plaintext, yang diubah menjadi kode-kode yag tidak dimengerti. Enkripsi dapat diartikan dengan *cipher* atau kode. Sama halnya dengan kita tidak mengerti akan sebuah kata maka kita akan melihatnya didalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah text asli kebentuk text kode kita menggunakan algoritma yang dapat mengkodekan data yang kita inginkan.

2. Dekripsi: merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan kebentuk asalnya (text asli), disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi.

3. Kunci: yang dimaksud disini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).



Gambar 2.1 Proses Enkripsi dan Deskripsi Pesan.

2.1.1 Jenis Kriptografi Berdasarkan Perkembangan

Berdasarkan perkembangannya kriptografi dapat diklasifikasikan menjadi 2 jenis yaitu kriptografi klasik dan kriptografi modern.

a. Kriptografi Klasik

Kriptografi klasik digunakan sejak sebelum era komputerisasi ada, kebanyakan manusia dulu menggunakan kunci simetris. Teknik kriptografi simetris yang digunakan untuk mengacak pesan adalah teknik substitusi dan

transposisi atau bahkan menggabungkan kedua teknik tersebut. Teknik substitusi adalah teknik untuk menggantikan suatu karakter dalam pesan asli menjadi

karakter lain yang hasilnya adalah pesan teracak yang telah dienkripsi. Sedangkan teknik transposisi yaitu teknik mengubah pesan asli menjadi pesan teracak yang telah dienkripsi dengan cara permutasi karakter. Penggabungan kedua teknik tersebut dapat menghasilkan atau membentuk berbagai macam kriptografi klasik lainnya.

b. Kriptografi Modern

Kriptografi modern merupakan perkembangan dari kriptografi klasik. Algoritma ini memiliki tingkat kesulitan yang lebih sulit dibandingkan dengan algoritma kriptografi klasik, dan kekuatan dari pengacakan pesan terdapat pada kuncinya. Algoritma kriptografi modern menggunakan pengolahan simbol biner karena berjalan mengikuti operasi komputer digital. Sehingga membutuhkan pengetahuan terhadap matematika untuk menguasainya.

2.1.2. Jenis kriptografi Berdasarkan Kunci

Kriptografi dikelompokkan menjadi 2 jenis berdasarkan kuncinya, yakni simetris dan asimetris.

a. Kriptografi simetris

simetris bersifat sebagai algoritma tunggal, karena kuncinya atau sandi hanya untuk sipembuat dan orang yang akan menerima pesan tersebut. Proses enkripsi dan deskripsi algoritma ini sangat rahasia sehingga kunci atau sandi tersebut tidak akan dapat diketahui oleh umum.

b. Kriptografi asimetris

Berbeda dengan algoritma simetris, algoritma asimetris bersifat umum, karena dapat dikirim ke satu orang bahkan ke publik. Karena proses enkripsi kunci ini diperuntukkan untuk umum, sehingga pihak ketiga dapat mengetahui isi pesan yang akan disampaikan. Sedangkan untk proses deskripsi, kuncinya bersifat rahasia, sehingga kunci atau sandi tersebut hanya diketahui oleh orang tertentu. Dengan kata lain hanya orang yang dipercaya oleh si pembuat yang dapat memecahkan kode-kode enkripsinya kunci tersebut.

2.2 Playfair Cipher

Playfair Cipher merupakan suatu diagram *cipher* substitusi yang ditemukan oleh *Charles Wheatstone* namun dipromosikan oleh *Baron Lyon Playfair* pada tahun 1854 dan pertama kali digunakan oleh bangsa Inggris (Stinson, 1995). Cipher ini mengenkripsi pasangan karakter (bigram atau digraf) bukan karaker tunggal seperti pada cipher klasik lainnya. Tujuannya adalah untuk membuat analisis frekuensi menjadi sangat sulit sebab frekuensi kemunculan karakter-karakter didalam *ciphertext* menjadi datar atau *flat* (Sasongko, 2005).

Pada proses penyandian matriks kunci akan diisi sesuai dengan urutan kemunculan huruf pada kunci. Huruf yang digunakan tidak boleh digunakan lagi, sedangkan huruf yang tidak digunakan kunci akan disusun setelahnya sesuai dengan urutan alphabet. Menurut *Stallings* Sebelum melakukan enkripsi, pesan yang akan dienkripsi (*plaintext*) diatur terlebih dahulu sebagai berikut :

1. Semua spasi dan karakter yang bukan alfabet harus dihilangkan dari *plaintext* (jika ada).
2. Jika ada huruf J pada *plaintext*, maka ganti huruf tersebut dengan huruf I.
3. Pesan yang akan dienkripsi ditulis dalam pasangan huruf (bigram).
4. Jika ada huruf yang sama dalam pasangan huruf, maka sisipkan huruf X atau Z di tengahnya. Huruf yang disisipkan sebaiknya huruf X karena sangat kecil kemungkinan terdapat huruf X yang sama dalam bigram, tidak seperti huruf Z

5. Jika jumlah huruf pada *plaintext* adalah ganjil maka pilih sebuah huruf tambahan yang dipilih oleh orang yang mengenkripsi dan tambahkan di akhir *plaintext*. Huruf tambahan dapat dipilih sembarang misalnya huruf Z atau X.

2.2.1 Algoritma Playfair Cipher

a. Algoritma enkripsi untuk setiap bigram adalah sebagai berikut:

1. Jika ada dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kanannya.
2. Jika ada dua huruf terdapat pada kolom yang sama maka tiap huruf diganti dengan huruf di bawahnya

3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua.
4. Huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari huruf yang digunakan.

b. Algoritma deskripsi merupakan kebalikan dari algoritma enkripsi untuk setiap bigram adalah sebagai berikut:

1. Jika ada dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kirinya
2. Jika ada dua huruf terdapat pada kolom yang sama maka tiap huruf diganti dengan huruf di atasnya.
3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua.
4. Huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari huruf yang digunakan.

Proses enkripsi pada metode *playfair cipher* dilakukan dengan cara berikut :

Langkah pertama : Membuat matriks kunci, yaitu dengan memasukkan kunci kedalam matriks 5x5 sebagai berikut :

Kunci : SEHAT

Gambar 2.2. Matriks 5x5 *Playfair Cipher*

Langkah kedua : Susun *plaintext* sesuai dengan bigram dan aturan :

Plaintext : HALAMAN SATU

Plaintext yang disusun sesuai bigram dan aturan : HA LA MA NS AT UX

1. HA LA menjadi AT MH
2. MA NS menjadi RF IT
3. AT UX menjadi TS QZ

Sehingga didapat ;

Plaintext: HA LA MA NS AT UX

Chipertext: AT MH RF IT TS QZ

2.3 Steganografi

Pada teknik kriptografi, data yang telah disandikan (*chiperteks*) tetap tersedia, maka dengan steganografi *cipherteks* dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya. Kata steganografi (*steganography*) berasal dari bahasa Yunani yaitu *steganos* yang berarti tersembunyi atau terselubung dan *graphia* yang artinya menulis, sehingga arti steganografi adalah "menulis (tulisan) terselubung" (Darmayanti, 2016).

Steganografi juga merupakan ilmu menyembunyikan teks pada media lain yang telah ada sedemikian sehingga teks yang tersembunyi menyatu dengan media itu. Menurut Cahyadi (2012), terdapat beberapa contoh media penyisipan pesan rahasia yang digunakan dalam teknik steganografi antara lain:

1. Teks

   Dalam algoritma steganografi yang menggunakan teks sebagai media penyisipan biasanya digunakan teknik NLP sehingga teks yang telah disisipkan pesan rahasia tidak akan mencurigakan untuk orang yang melihatnya.

2. Audio

   Format ini pun sering dipilih karena biasanya berkas dengan format ini berukuran relatif besar, sehingga dapat menampung pesan rahasia dalam jumlah yang besar pula.

3. Citra

   Format ini juga sering digunakan karena format ini merupakan salah satu format file yang sering dipertukarkan dalam dunia internet. Alasan lainnya

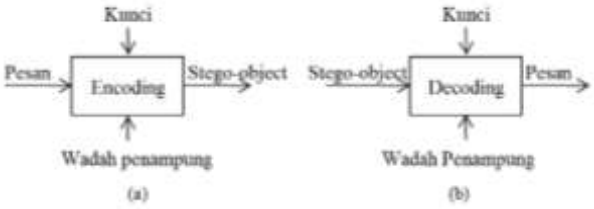
adalah banyaknya tersedia algoritma steganografi untuk media penampung yang berupa citra.

4. Video

Format ini memang merupakan format dengan ukuran file yang relatif sangat besar namun jarang digunakan karena ukurannya yang terlalu besar sehingga mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

2.3.1. Proses Steganografi

Secara umum, terdapat dua proses didalam steganografi, yaitu proses encoding untuk menyisipkan pesan ke dalam *cover – object* dan proses *decoding* untuk ekstraksi pesan dari *stego – object*. Kedua proses ini mungkin memerlukan kunci rahasia (*stegokey*) agar hanya pihak yang berhak saja yang dapat melakukan penyisipan pesan dan ekstraksi pesan (Munir, 2006).



Gambar 2.3 (a) Skema *Encoding* (b)Skema *Decoding*

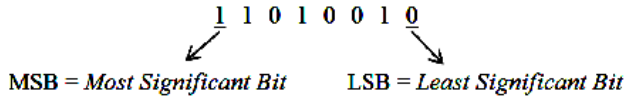
Menurut Munir (2004) dalam melakukan steganografi terdapat beberapa kriteria yang harus diperhatikan adalah :

1. *Fidelity* mutu media penampung tidak jauh berubah, setelah penambahan data rahasia, *stego object* dalam kondisi yang masih terlihat baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2. *Robustness* data rahasia yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi atau editing pada media penampung.
3. *Recovery* data yang disembunyikan harus dapat di ungkapkan kembali, karena dikaitkan dengan tujuan dari steganografi digital itu sendiri yaitu

data sewaktu-waktu data rahasia di dalam media penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

2.4 Least Significant Bit (LSB)

Metode *Least Significant Bit* (LSB) merupakan metode steganografi yang paling sederhana dan paling mudah diimplementasikan. LSB merupakan teknik penyembunyian pesan pada lokasi bit terendah dalam citra digital. Pesan rahasia akan dikonversikan ke dalam bentuk biner dan disembunyikan kedalam sebuah media penyembunyian berupa citra digital. Hasil dari penyembunyian pesan menggunakan metode LSB tidak memiliki perubahan sehingga sulit dibedakan oleh mata manusia. Untuk menjelaskan metode LSB ini kita menggunakan citra digital sebagai *cover – object*. Setiap pixel di dalam citra berukuran 1 sampai 3 *byte*. Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit yang paling berarti yaitu MSB (*most significant bit*) dan bit yang paling kurang berarti yaitu LSB (*least significant bit*).



Gambar 2.4. Contoh MSB dan LSB

Dari contoh *byte* pada Gambar 2.4, bit 1 yang pertama (digaris bawahi) adalah MSB dan bit 0 yang terakhir (digaris bawahi) adalah LSB. Bit yang cocok untuk diganti dengan bit pesan adalah LSB, karena modifikasi hanya menggunakan nilai *byte* tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalnya *byte* tersebut di dalam citra memberikan persepsi warna merah, maka perubahan satu bit LSB hanya mengubah persepsi warna merah tidak terlalu berarti. Mata manusia tidak dapat membedakan perubahan sekecil ini. Sebagai ilustrasi, misalkan segmen pixel - pixel citra sebelum disisipkan pesan adalah:

00110011 10100010 11100010 01101111

Dan misalkan pesan rahasia (yang telah dikonversi ke biner) adalah 0110. Setiap bit pesan menggantikan posisi LSB dari segmen pixel – pixel citra menjadi:

00110010 10100011 11100011 10010000

2.5 Pengertian Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra yang berupa keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan (Sutoyo, 2009).

2.5.1. Citra Analog

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak dikertas foto, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung. Agar citra ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu (Sutoyo, 2009).

2.5.2. Citra Digital

Citra digital adalah citra yang bersifat diskrit yang dapat diolah oleh komputer yang merupakan suatu *array* dari bilangan yang merepresentasikan intensitas terang pada point yang bervariasi (pixel). Citra ini dapat dihasilkan melalui kamera digital dan scanner ataupun citra yang telah mengalami proses digitalisasi. Citra digital disimpan juga secara khusus di dalam file 24 bit atau 8 bit. Citra 24 bit menyediakan lebih banyak ruang untuk menyembunyikan informasi (Sutoyo, 2009).

2.5.3. Jenis-Jenis Citra Digital

Citra digital dapat dibagi berdasarkan warna – warna penyusunannyan menjadi tiga macam (Wildan, 2010) yaitu:

1. Citra Biner

Citra biner adalah citra yang hanya memiliki 2 warna, yaitu hitam dan putih. Oleh karena itu, setiap pixel pada citra biner cukup direpresentasikan

dengan 1 bit. Alasan penggunaan citra biner adalah karena citra biner memiliki sejumlah keuntungan sebagai berikut:

a. Kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit.
b. Waktu pemrosesan lebih cepat di bandingkan dengan citra hitam – putih ataupun warna.

2. Citra Grayscale

Citra grayscale adalah citra yang nilai pixel-nya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Pada umumnya citra *grayscale* memiliki kedalaman pixel 8 bit (256 derajat keabuan), tetapi ada juga citra *grayscale* yang kedalaman pixel-nya bukan 8 bit, misalnya 16 bit untuk penggunaan yang memerlukan ketelitian tinggi.

3. Citra Warna

Citra warna adalah citra yang nilai pixel-nya merepresentasikan warna tertentu. Setiap pixel pada citra warna memiliki warna yang merupakan kombinasi 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 pixel mempunyai kombinasi warna sebanyak $2^8.2^8.2^8 = 2^{24}$ = 16 juta warna lebih. Itulah yang menjadikan alasan format ini disebut dengan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

2.6 Pengertian Website

Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.7 Pengertian PHP

PHP sendiri sebenarnya merupakan singkatan dari (*Personal Home Page*). Selanjutnya diganti menjadi FI (*Forms Interpreter*). Sejak versi 3.0, nama bahasa ini diubah menjadi *" PHP : Hypertext Preprocessor "*, yang merupakan sebuah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan biasanya digunakan bersamaan dengan HTML. Sebagian besar sintaks dalam PHP mirip dengan bahasa C, Java dan Perl, namun pada PHP ada beberapa fungsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancangan web yang dinamis dan dapat bekerja secara otomatis. Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML dan CSS ini dikenal dengan website statis, dimana konten didalam web bersifat tetap. Sebagai perbandingan, website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari *user*, memproses form, dll. Untuk pembuatan web, kode PHP biasanya disisipkan ke dalam dokumen HTML.Karena fitur inilah PHP disebut juga sebagai *scripting language* atau bahasa pemrograman *script*.

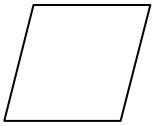
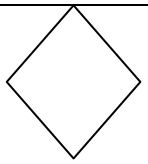
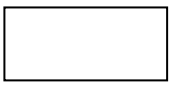
2.8 Flowchart

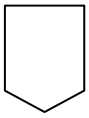
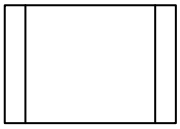
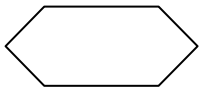
Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer. Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk

menghubungkan satu proses ke proses selanjutnya digunakan dengan menggunakan garis penghubung.

Dengan adanya *flowchart*, setiap urutan proses dapat digambarkan menjadi lebih jelas. Selain itu ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan *flowchart* ini. Setelah proses membuat *flowchart* selesai, maka giliran programmernya yang akan menerjemahkan desain logis tersebut kedalam bentuk program dengan berbagai bahasa pemrograman.

Tabel 2.1 Flowchart

Simbol	Keterangan
	**Terminal** (STAR, END atau Mulai, Selesai)
	**Input/Output** Merepresentasikan Input data atau Output data yang di proses atau Informasi.
	**Decision** Keputusan dalam program
	**Processing** Proses (menyatakan assignment statement)
	**Display** Output yang ditampilkan pada terminal
	Connecting Line Untuk menghubungkan simbol satu dengan yang lainnya , meyatakan arus suatu proses

	**Connector** Keluar ke atau masuk dari bagian lain *flowchart* khususnya halaman yang sama
	**Offline Connector** Untuk menyatakan sambungan dari proses yang satu ke proses berikutnya dihalaman yang berbeda
	**Predefined Process** Rincian operasi berada di tempat lain
	**Disk Storage** I/O yang menggunakan penyimpanan akses langsung
	**Document** I/O yang berasal dari dokumen
	**Preparation** Pemberian nilai awal suatu variabel

2.9 Kode ASCII

Kode ASCII(American Standard Codes for International Interchange) adalah kumpulan kode-kode yang digunakan untuk mempermudah interaksi antara user dan komputer. “Interaksi” yang dimaksud adalah sarana untuk menyelesaikan permasalahan hubungan antara komputer yang hanya mengenal angka, sedangkan manusia tidak mungkin harus menghafal angka yang cukup banyak tersebut dan menggunakan keyboard sebagai masukan atas perintah yang diinginkannya. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai

dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan Desimal. Pada dasarnya kode ASCII merepresentasikan kode-kode untuk :

1. Angka (0,1,2,3,4,5,6,7,8,9)
2. Huruf (a –z , A - Z)
3. Simbol (&, ^, %, $, ….)
4. Tombol (Enter, Esc, Tab, ….)
5. Karakter Grafis (kode ASCII Standar nomor 128 s/d 255)
6. Kode Komunikasi (ETX, STX, ENQ,…)

A. Kode Standard ASCII

Kode ini merepresentasikan angka, huruf serta tombol standar, Enter, Escape, Backspace dan Space. Selain itu juga terdapat karakter-karakter yang tidak terdapat pada keyboard, yang dapat diaktifkan dengan melakukan penekanan tombol kombinasi “Alt” dan angka yang dimaksud, sebagai contoh tombol kombinasi “Alt” dan angka “127″ akan menghasilkan karakter grafis. Karakter dasar lain juga digunakan untuk komunikasi, seperti yang Anda ketahui bersama, karakter tersebut adalah “ACK” dan “ENQ”. Pada saat akan dilakukan komunikasi pada jaringan dengan protokol Ethernet, maka bentuk komunikasi yang terjadi adalah komputer akan mengirimkan “ACK” (Acknowledge) pada computer lain yang akan berkomunikasi, jika komputer lain merespon, maka komputer tersebut akan membalasnya dengan mengirim “ENQ” (Enquiry).

B. Kode Extended ASCII

Kode ASCII Extended akan bertindak sebagai kode perluasan (extended) dari kode ASCII yang ada, karena tidak semuanya mampu tertampung dalam kode ASCII standard. Kode ASCII jenis ini lebih banyak bertindak sebagai kode-kode tombol khusus, seperti kode untuk tombol F1 s/d F12. Sebagai contoh adalah kode ASCII extended untuk F12 adalah “123″. Belum lagi dengan tombol kombinasi, misalnya “Alt” dan “F1″, “Ctrl” dan “F1″, atau tombol-tombol yang biasa kita lakukan “Alt” + “F” untuk membuka menu file, “Ctrl” dan “O” untuk membuka dokumen dsb.

BAB III
METODOLOGI PENELITIAN

3.1 Tempat dan Waktu

3.1.1 Tempat Penelitian

Penelitian ini dilakukan dilakukan Laboratorium Jurusan Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri sumatera Utara, yang beralamat di Jalan IAIN No.01 Medan.

3.1.2 Waktu Penelitian

Waktu penelitian dilakukan pada semester genap 2020/2021 dimulai dari bulan April 2020 .

3.2 Bahan dan Alat

Bahan yang digunakan dalam penelitian ini adalah sebuah image atau gambar yang berekstensi *.jpg .

3.2.1 Perangkat Keras

Adapun Perangkat Keras yang dibutuhkan dalam pembuatan sistem adalah laptop Dell dengan spesifikasi Proseccor Intel(R) Celeron(R) CPU N3350 @1.10GHz 1.10GHz, Memori 4,00 GB 64bit.

3.2.2 Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan dalam proses pembuatan sistem adalah sebagai berikut:

1. Sistem Operasi Windows 10 pro dari *Microsoft Corporation 2018*
2. Xampp v.3.2.2
3. Sublime Text 3

3.3 Prosedur Kerja

Pada penelitian ini terlebih dahulu mempelajari materi-materi dasar untuk penelitian ini seperti kriptografi, steganografi, serta mempelajari pembuatan web menggunakan PHP. Selanjutnya melakukan enkripsi dan proses penyisipan pesan

kedalam sebuah gambar lalu melakukan proses deskripsi dan ekstraksi untuk mendapatkan pesan asli.

3.3.1 Perencanaan

Tahap awal yang dilakukan adalah pendefenisian masalah yang akan diselesaikan dari sistem yang akan dibangun yaitu bagaimana membangun sistem meningkatkan keamanan pada pesan rahasia dengan mengubahnya kedalam bentuk yang tidak bermakna bagi pihak yang tidak bertanggung jawab dan menyisipkannya kedalam sebuah media sehingga tingkat pengamanan lebih tinggi dan tidak diketahui oleh pihak ketiga. Maka akan dibangun suatu sistem pengamanan pesan rahasia dengan menggunakan kriptografi Playfair Cipher dan menyisipkan pesan rahasia yang telah dienkripsi tersebut kedalam media gambar dengan metode Least Significant Bit (LSB).

3.3.2 Teknik Pengumpulan Data

Pada bab ini penulis menjelaskan menguraikan teknik penelitian yang digunakan dalam perancangan sistem yaitu studi literatur. Penulis membaca buku-buku dan jurnal-jurnal yang berkaitan dengan kriptografi, steganografi dan pengolahan citra. Tujuan dari studi literatur adalah untuk memperoleh sumberreferensi yang berisi teori-teori serta penelitian-penelitian yang telah dilakukan sebelumnya guna memberikan kemudahan dalam melakukan penelitian ini.

3.3.3 Analisis Kebutuhan

A. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional mendeskripsikan fungsi-fungsi yang harus dilakukan oleh sebuah sistem untuk mencapai tujuan. Kebutuhan fungsional yang harus dimiliki dalam sistem adalah sebagai berikut:

1. Sistem melakukan enkripsi pesan *Plaintext* yang diinputkan oleh *user* atau mengambil file yang telah ada dalam bentuk **.txt* dengan kunci yang juga diinputkan oleh *user*. Proses enkripsi, sistem akan menghasilkan *ciphertext* yang kemudian akan digunakan untuk proses penyisipan kedalam gambar.

2. Sistem akan menyisipkan pesan kedalam media gambar yang diinputkan *user* berformat **.jpg*. Sistem kemudian menyimpan hasil gambar yang telah disisipikan pesan didalamnya.

3. Pada proses ektraksi sistem melakukan ekstraksi pada gambar yang diinputkan *user* untuk mengambil pesan yang masih berupa *ciphertext* .

4. Sistem mendeskripsikan *ciphertext* yang didapat dari proses ekstraksi sehingga menghasilkan *plaintext*. Sistem dapat menampilkan hasil deskripsi berupa *plaintext*.

B. Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional dapat digunakan sebagai suatu bentuk kebutuhan berupa perangkat yang dibutuhkan sistem. Analisa kebutuhan yang digunakan dalam pengembangan sistem ini yaitu perangkat keras berupa laptop yang berspesifikasi sebagai berikut: *Proseccor* Intel(R) Celeron(R) CPU N3350 @1.10GHz 1.10GHz, Memori 4096MB RAM DDR3L. directX *version*: DirectX 12, Chip *type*: Intel(R) HD *Graphics Family*.

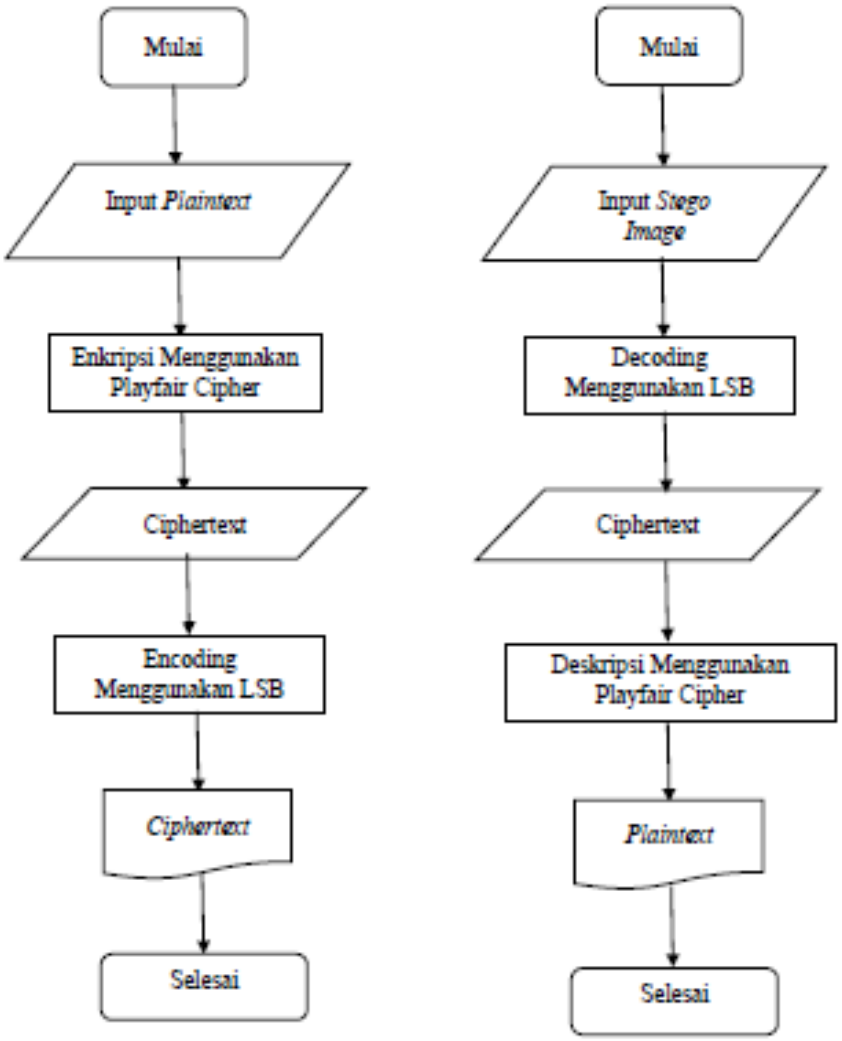
Perangkat lunak atau software yang digunakan yaitu sistem operasi Windows 10, XAMPP v3.2.1, Sublime Text3, dan browser Google Chrome.

3.3.4 Perancangan

Perancangan sistem merupakan proses penggambaran dan pembuatan alur kerja sistem, serta gambaran-gambaran dari bentuk sistem memasuki tahap implementasi. Berikut merupakan penjabaran dari bentuk sistem yang dibangun yang akan ditampilkan dalam bentuk *flowchart* .

A. Flowchart Sistem

Secara garis besar penelitian ini terdari dari dua proses, yaitu proses enkripsi dan proses *encoding*. Pada proses enkripsi menggunakan algoritma Playfair Cipher, sedangkan pada proses *encoding* menggunakan metode steganografi *Least Significant Bit*. Kemudian penulis juga akan melengkapi penelitian ini dengan proses *decoding* dan deskripsi agar pesan yang telah dienkripsi dan disembunyikan dapat kembali menjadi pesan asli yang dapat digunakan.

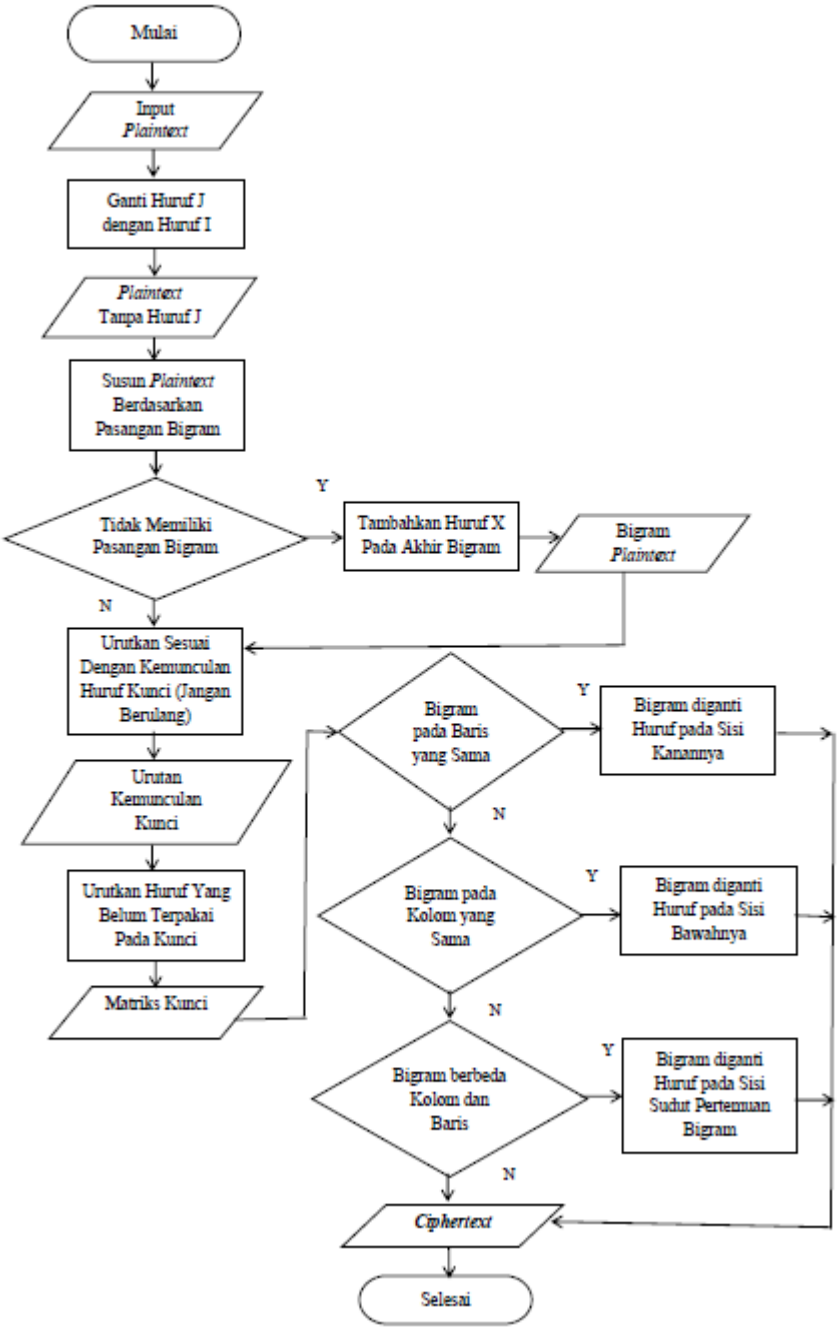


(a) **(b)**

Gambar 3.1 (a) Flowchart enkripsi dan *encoding*; (b) Flowchart *decoding* dan Deskripsi

B. *Flowchart* Proses Enkripsi

Proses enkripsi pada sistem ini menggunakan *Playfair Cipher*. Berikut ini *flowchart* yang menggambarkan langkah-langkah proses pengenkripsian pesan.



Gambar 3.2 *Flowchart* Untuk Proses Enkripsi

Dari *flowchart* pada gambar dapat kita ketahui proses enkripsi pada tahap awal, user menginputkan plainteks yang kemudian akan di enkripsi dengan menggunakan *playfair cipher*. Untuk implementasi dari proses enkripsi pesan, maka dilakukan langkah berikut :

1. User menginputkan *Plainteks* : **COBA1**.

2. Apabila plainteks terdapat huruf **J** maka akan diganti dengan huruf **I**.
3. Plainteks akan disusun menjadi bigram : **CO BA 1.**
4. Apabila huruf tidak memiliki pasangan bigram maka ditambahkan huruf ‘**X’** pada akhir bigram. Plainteks menjadi **CO BA 1X**.
5. Inputkan Kunci: **TES**
6. Urutkan sesuai dengan kemunculan huruf kunci , Kemudian urutkan huruf belum terpakai pada kunci. Sehingga menghasilkan matriks kunci sebagai berikut:

   Proses Enkripsi menggunakan *Playfair Cipher* dengan matriks kunci 6x6.

T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.3 Matriks Kunci “TES”

7. Lakukan enkripsi menggunakan aturan algoritma Playfair Cipher dari *plainteks* “CO”. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua. Dan pada proses enkripsi “CO” akan menghasilkan *cipherteks* “QA” .

T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.4 Matriks Enkripsi “CO”

8. Lakukan enkripsi dari *plainteks* “BA” Jika ada dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kanannya. Dan menghasilkan *cipherteks* “CB”

T	E	S	**A**	**B**	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

T	E	S	A	**B**	**C**
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.5 Matriks Enkripsi “BA”

9. Lakukan enkripsi dari *plainteks* “1X” dan menghasilkan *cipherteks* “2W”

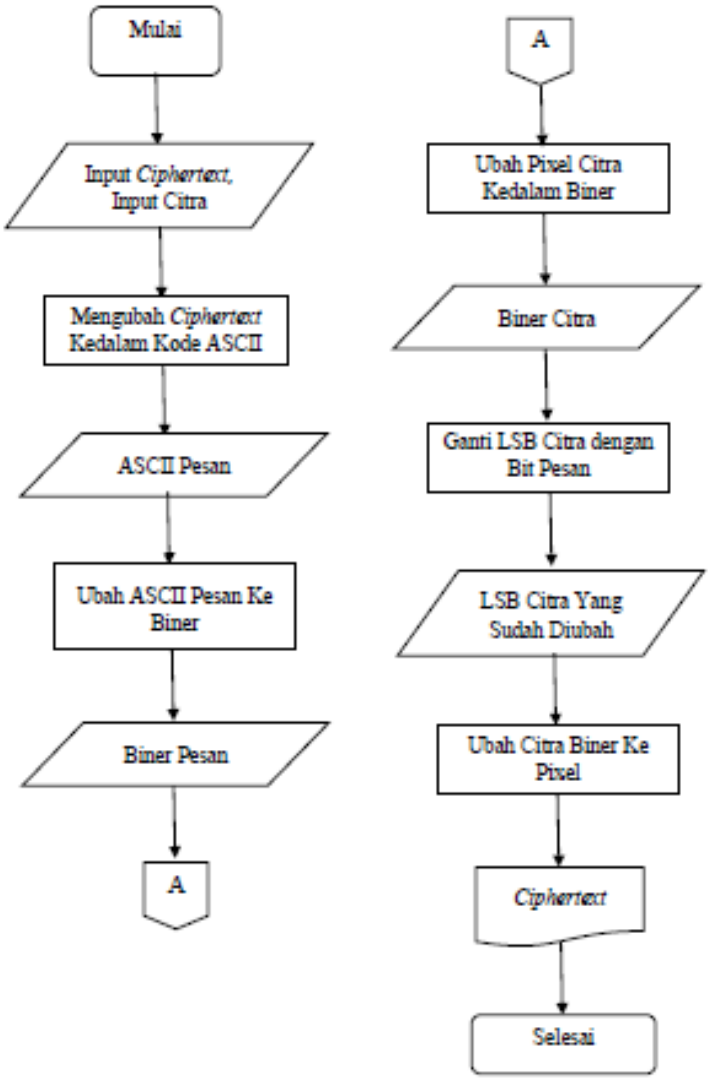
T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.6 Matriks Enkripsi “1X”

10. Setelah melakukan enkripsi terhadap seluruh *plainteks* “ **COBA1** “, maka diperoleh *cipherteks* “ **QACB2W** “.

C. *Flowchart* Proses Penyisipan Pesan (*Encoding*)

Proses penyisipan pesan kedalam sebuah gambar menggunakan metode *Least Significant Bit*. LSB mengganti setiap ujung bit citra dengan bit pesan yang akan disisipkan.



Gambar 3.7 *Flowchart* Proses Penyisipan Pesan (*Encoding*)

Flowchart diatas yang menggambarkan langkah-langkah proses pengenkripsian pesan. Berikut implementasi dari metode ini adalah sebagai berikut:

1. Dari gambar diatas dapat dilihat, cipherteks yang sebelumnya didapat dari proses enkripsi di inputkan.
2. Setelah diperoleh hasil cipherteks, maka setiap karakter diubah kebentuk kode ASCII dan diubah kedalam bentuk biner agar dapat diimplementasikan ke *cover image* yang digunakan

3. Pesan yang akan disisipkan adalah "**QACB2W**" pesan tersebut harus di representasikan kedalam biner sehingga didapat :

Tabel 3.1 Tabel Perubahan Dari Setiap Karakter

Karakter	ASCII	Hexadecimal	Binary
Q	81	51	01010001
A	65	41	01000001
C	67	43	01000011
B	66	42	01000010
2	50	32	00110010
W	87	57	01010111

4. Kemudian siapkan citra gambar yang akan menjadi wadah penampung pesan rahasia, kemudian ubah pixel citra ke biner dan menghasilkan sebagai berikut.

Tabel 3.2 Citra Dalam Bentuk Biner

00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011

5. Siapkan Data cipherteks yang setiap karakternya telah diubah kedalam bentuk biner dan akan disisipkan kedalam bit gambar :

Tabel 3.3 Data *Ciphertext* Yang Akan Disisipkan

0	1	0	1	0	0	0	1
0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	1
0	1	0	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	1	0	1	1	1

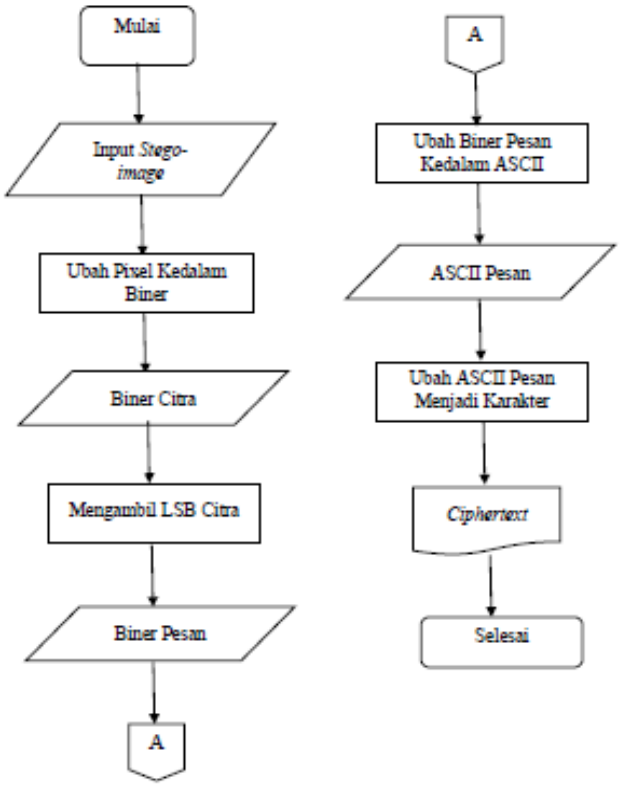
6. Melakukan Proses penyisipan dengan mengganti bit terakhir dari gambar dengan bit dari setiap karakter *ciphertext*. Dari proses penyisipan karakter *ciphertext* di peroleh s*tego image* seperti tabel dibawah. Perubahan piksel-piksel gambar hanya terjadi pada bit-bit paling belakang dari *stego image.* Dengan perubahan yang tidak signifikan tidak akan terdeteksi oleh mata manusia sehingga tidak akan mengandung kecurigaan.

Tabel 3.4 Stego Image

00000000	00000001	00000000	00000001	00000000	00000000	00000000	00000001
00000000	00000001	00000000	00000000	00000000	00000000	00000000	00000001
00000000	00000001	00000000	00000000	00000000	00000000	00000001	00000001
00000000	00000001	00000010	00000010	00000010	00000010	00000011	00000010
00000000	00000001	00000000	00000001	00000000	00000000	00000000	00000001
00000000	00000000	00000011	00000011	00000010	00000010	00000011	00000010
00000000	00000001	00000010	00000011	00000010	00000011	00000011	00000011

D. *Flowchart* Proses Ekstraksi Pesan (*Decoding*)

Proses ektraksi merupakan cara untuk mengambil pesan yang disisipkan ke dalam gambar. Langkah –langkah ekstraksi pesan dapat dilihat pada gambar dibawah ini.



Gambar 3.8 *Flowchart* Proses Ekstraksi Pesan (*Decoding*)

Berikut implementasi dari proses ektraksi menggunakan *least significant bit* :

1. Mengubah nilai matriks citra s*tego image* kedalam bentuk biner . Kemudian mengambil bit terakhir dari setiap piksel , nilai biner tersebut merupakan biner pesan yang telah disisipkan kedalam citra.

Tabel 3.5 Matriks Citra Dalam Bentuk Biner

00000000	00000001	00000000	00000001	00000000	00000000	00000000	00000001
00000000	00000001	00000000	00000000	00000000	00000000	00000000	00000001
00000000	00000001	00000000	00000000	00000000	00000000	00000001	00000001
00000000	00000001	00000010	00000010	00000010	00000010	00000011	00000010
00000000	00000000	00000011	00000011	00000010	00000010	00000011	00000010
00000000	00000001	00000010	00000011	00000010	00000011	00000011	00000011

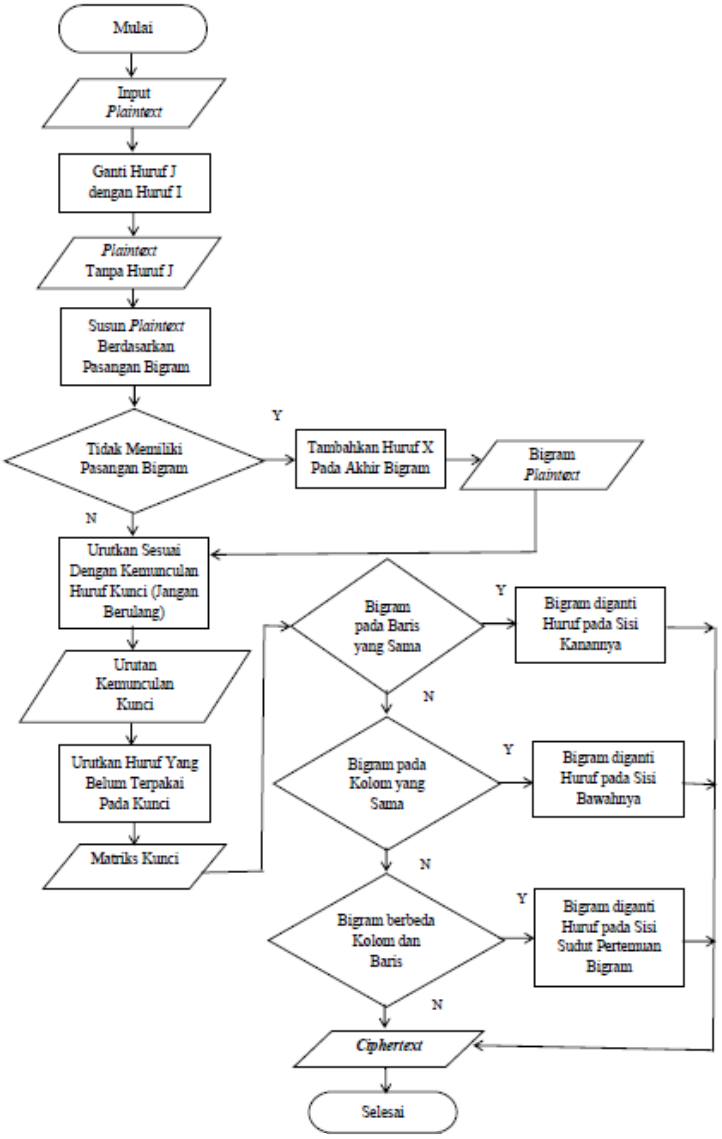
2. Mengubah nilai biner pesan yang disisipkan kedalam bentuk desimal agar dapat diketahui pesan yang dimaksud. Nilai desimal inilah yang merupakan nilai bentuk ASCII dari pesan tersebut. Maka dari tabel dibawah dapat diketahui bahwa pesan yang disisipkan yaitu **“ QACB2W “** pesan ini disebut *cipherteks.*

Tabel 3.6 Mengubah Biner Ke Bentuk ASCII

Binary	ASCII	Karakter
01010001	81	Q
01000001	65	A
01000011	67	C
01000010	66	B
00110010	50	2
01010111	87	W

E. *Flowchart* Proses Deskripsi

Proses deskripsi dilakukan setelah berhasil didapatnya *cipherteks* dari proses ekstraksi gambar. Langkah-langkah proses deskripsi dapat dilihat pada flowchart di gambar 3.9.



Gambar 3.9 *Flowchart* Proses Deskripsi

Deskripsi pesan menggunakan *playfair cipher* merupakan kebalikan dari proses enkripsi, berikut adalah implementasi dari proses deskripsi:

1. Siapkan *Ciphertext* yaitu “QACB2W”
2. Lakukan deskripsi menggunakan aturan algoritma *Playfair Cipher* dari *ciphertext* “QA” Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua dan akan menghasilkan “CO”.

T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.10 Matriks Deskripsi "QA"

3. Lakukan deskripsi menggunakan algoritma *Playfair Cipher* dari *ciphertext* "CB" menjadi "BA".

T	E	S	**A**	**B**	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

T	E	S	A	**B**	**C**
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.11 Matriks Deskripsi "BC"

4. Lakukan deskripsi menggunakan algoritma *Playfair Cipher* dari *ciphertext* "2W" menjadi "1X"

T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 3.12 Matriks Enkripsi "2W"

5. Kemudian hasil dari enkripsi tersebut merupakan *plaintext* ataupun pesan asli yaitu COBA1X.

3.3.5 Pengujian

Tahap pengujian adalah tahap untuk memastikan seluruh kebutuhan telah diimplementasikan bekerja dengan semestinya serta mengidentifikasi kekurangan dari pada sistem. Pada tahap ini terdapat beberapa hal yang akan dilakukan dalam pengujian yaitu:

A. Pengujian Enkripsi dan Deskripsi

Pengujian ini dilakukan untuk membuktikan apakah proses enkripsi pesan rahasia dapat diubah kedalam bentuk yang tidak dapat dimengerti oleh pihak ketiga. Dan sebaliknya pada saat melakukan deskripsi, apakah pesan yang tidak dapat dimengerti tersebut dapat dikembalikan kedalam bentuk yang dapat dimengerti sesuai pesan aslinya tanpa mengurangi, menambahkan, dan memodifikasi isinya.

B. Pengujian Pengiriman *Stego-image* Melalui Sosial Media

Pengujian ini dilakukan untuk membuktikan apakah pengiriman *Stego-image* melalui jalur komunikasi pada beberapa aplikasi sosial media dapat sampai dengan utuh tanpa mengalami kerusakan berkas. Pengujian ini akan dilakukan dengan cara mengirimkan *stego-image* ke beberapa aplikasi sosial media seperti *Line, Whatsapp, Telegram* dan *facebook Messenger*

3.3.6 Penerapan/Penggunaan

Penerapan penelitian ini yaitu menerapkan kriptografi dalam penyisipan pesan pada citra gambar dengan menggunakan kriptografi algoritma *Playfair Cipher* dan metode steganografi *Least Significant Bit* (LSB). Hal ini dilakukan untuk mendapatkan tingkat keamanan yang lebih tinggi guna melindungi pesan rahasia tetap aman. Dalam implementasi ini, pesan rahasia yang dienkripsi berupa pesan teks yang dibuat secara manual atau file berformat (**.txt*). sedangkan citra yang digunakan sebagai wadah penyisipan pesan menggunakan format file (**.jpg*). Proses ekstraksi berkas dengan cara memasukkan *stego-image* yang dihasilkan dari proses penyisipan guna mendapatkan *ciphertext* dan selanjutnya *ciphertext* dideskripsi agar mendapatkan pesan rahasia atau *plaintext*.

Implementasi sistem ini dibangun menggunakan bahasa pemrograman PHP yang diaplikasikan berbasis web.

BAB IV
HASIL DAN PEMBAHASAN

4.1. Hasil

Untuk menjelaskan hasil dari sistem yang dirancang, maka penjabaran akan dibagi menjadi tiga bagian, yaitu perhitungan manual, gambaran *flowchart* dan tampilan hasil antarmuka pemakai.

4.1.1. Perhitungan Manual

Proses perhitungan dari prosedur enkripsi dengan metode *Playfair Cipher* dapat dijabarkan sebagai berikut:

1. Pesan rahasia = **BACA**
2. Kunci rahasia = **TES**
3. Plainteks akan disusun menjadi bigram : **BA CA.**
4. Apabila huruf tidak memiliki pasangan bigram maka ditambahkan huruf ‘X’ pada akhir bigram. Plainteks menjadi **BA CA**.
5. Buat matriks kunci dengan ukuran 6 x 6, dengan cara urutkan sesuai dengan kemunculan huruf kunci, kemudian urutkan huruf belum terpakai pada kunci.

T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 4.1 Matriks Kunci “TES”

6. Lakukan enkripsi dari plainteks “BA” dan menghasilkan cipherteks “CB”.

T	E	S	**A**	**B**	C		T	E	S	A	**B**	**C**
D	F	G	H	I	K		D	F	G	H	I	K
L	M	N	O	P	Q		L	M	N	O	P	Q
R	U	V	W	X	Y		R	U	V	W	X	Y
Z	␣	0	1	2	3		Z	␣	0	1	2	3
4	5	6	7	8	9		4	5	6	7	8	9

Gambar 4.2 Matriks Enkripsi "BA"

7. Lakukan enkripsi dari plainteks "CA" dan menghasilkan cipherteks "TB".

T	E	S	**A**	B	**C**		**T**	E	S	A	**B**	C
D	F	G	H	I	K		D	F	G	H	I	K
L	M	N	O	P	Q		L	M	N	O	P	Q
R	U	V	W	X	Y		R	U	V	W	X	Y
Z	␣	0	1	2	3		Z	␣	0	1	2	3
4	5	6	7	8	9		4	5	6	7	8	9

Gambar 4.3 Matriks Enkripsi "CA"

8. *Ciphertext* yang diperoleh = CBTB.

Setelah itu, *ciphertext* akan disisipkan ke dalam citra sampul. Proses perhitungan dari penyisipan data dengan metode LSB dapat dirincikan sebagai berikut:

1. Input citra sampul.

Gambar 4.4 Citra Sampul

2. Baca warna elemen RGB dari setiap piksel pada citra sampul.

Tabel 4.1 Tabel Citra Sampul

216,13,46	239,107,49	61,44,1	114,120,244	186,146,138
94,182,178	208,137,12	140,252,83	113,4,165	15,127,152
249,25,181	246,232,108	171,101,106	191,72,242	239,151,85
151,193,14	254,246,72	91,1,23	242,210,201	193,39,144
81,145,192	61,116,35	150,168,203	129,198,122	176,108,120

3. Baris pertama (baris paling atas) akan digunakan untuk menyimpan jumlah karakter pada *ciphertext*. Jumlah karakter dibatasi maksimal 255 karakter, yang berarti bit panjang ciphertext yang akan disisipkan adalah sebesar 8 bit.
   Panjang *ciphertext* = 4 karakter. Nilai ini akan dikonversikan ke biner menjadi **0000 0100**.
   Sisipkan setiap bit ke dalam elemen warna RGB dari setiap piksel.
   Piksel (1, 1)
   216= 1101 1000 = 1101 100**0** **= 216**
   13 = 0000 1101 = 0000 110**0** **= 12**
   46 = 0010 1110 = 0010 111**0** **= 46**

   Piksel (1, 2)
   239= 1110 1111 = 1110 111**0** **= 238**
   107= 0110 1011 = 0110 101**0** **= 106**
   49 = 0011 0001 = 0011 000**1** **= 49**

   Piksel (1, 3)
   61 = 0011 1101 = 0011 110**0** **= 60**
   44 = 0010 1100 = 0010 110**0** **= 44**
   1 = 0000 0001 = 1

4. *Ciphertext* akan disisipkan mulai dari baris 2 dari citra sampul.
   Ciphertext yang akan disisipkan:

C = 67 = 0100 0011

B = 66 = 0100 0010

T = 84 = 0101 0100

B = 66 = 0100 0010

Bit yang akan disisipkan: **0100 0011 0100 0010 0101 0100 0100 0010**

Piksel (2, 1)

94 = 0101 1110 = 0101 111**0 = 94**

182 = 1011 0110 = 1011 011**1 = 183**

178 = 1011 0010 = 1011 001**0 = 178**

Piksel (2, 2)

208 = 1101 0000 = 1101 000**0 = 208**

137 = 1000 1001 = 1000 100**0 = 136**

12 = 0000 1100 = 0000 110**0 = 12**

Piksel (2, 3)

140 = 1000 1100 = 1000 110**1 = 141**

252 = 1111 1100 = 1111 110**1 = 253**

83 = 0101 0011 = 0101 001**0 = 82**

Piksel (2, 4)

113 = 0111 0001 = 0111 000**1 = 113**

4 = 0000 0100 = 0000 010**0 = 4**

165 = 1010 0101 = 1010 010**0 = 164**

Piksel (2, 5)

15 = 0000 1111 = 0000 111**0 = 14**

127 = 0111 1111 = 0111 111**0 = 126**

152 = 1001 1000 = 1001 100**1 = 153**

Piksel (3, 1)

249 = 1111 1001 = 1111 100**0 = 248**

25 = 0001 1001 = 0001 100**0 = 24**

181 = 1011 0101 = 1011 010**1 = 181**

Piksel (3, 2)

246 = 1111 0110 = 1111 011**0 = 246**

232 = 1110 1000 = 1110 100**1 = 233**

108 = 0110 1100 = 0110 110**0 = 108**

Piksel (3, 3)

171 = 1010 1011 = 1010 101**1 = 171**

101 = 0110 0101 = 0110 010**0 = 100**

106 = 0110 1010 = 0110 101**0 = 106**

Piksel (3, 4)

191 = 1011 1111 = 1011 111**0 = 190**

72 = 0100 1000 = 0100 100**1 = 73**

242 = 1111 0010 = 1111 001**0 = 242**

Piksel (3, 5)

239 = 1110 1111 = 1110 111**0 = 238**

151 = 1001 0111 = 1001 011**0 = 150**

85 = 0101 0101 = 0101 010**0 = 84**

Piksel (4, 1)

151 = 1001 0111 = 1001 011**1 = 151**

193 = 1100 0001 = 1100 000**0 = 192**

14 = 0000 1110 = 0000 1110 = 14

5. Tampilkan citra stego yang diperoleh

Gambar 4.5 Citra Stego

Tabel 4.2 Citra Sampul Yang Telah Disisipi Pesan

216,12,46	**238,106,49**	**60,44**,1	114,120,244	186,146,138
94,183,178	**208,136,12**	**141,253,82**	**113,4,164**	**14,126,153**
248,24,181	**246,233,108**	**171,100,106**	**190,73,242**	**238,150,84**
151,192,14	254,246,72	91,1,23	242,210,201	193,39,144
81,145,192	61,116,35	150,168,203	129,198,122	176,108,120

Setelah diperoleh citra stego, maka file citra stego ini akan dikirimkan kepada penerima. Kemudian, penerima akan mengekstraksi data dari citra stego tersebut. Proses kerja dari ekstraksi data dari citra stego dapat dirincikan sebagai berikut:

1. Input citra stego

Gambar 4.6 Citra Stego Yang Akan Diekstraksi

Tabel 4.3 Citra Sampul Yang Akan Diekstraksi

216,12,46	**238,106,49**	**60,44**,1	114,120,244	186,146,138
94,183,178	**208,136,12**	**141,253,82**	**113,4,164**	**14,126,153**
248,24,181	**246,233,108**	**171,100,106**	**190,73,242**	**238,150,84**
151,192,14	254,246,72	91,1,23	242,210,201	193,39,144
81,145,192	61,116,35	150,168,203	129,198,122	176,108,120

2. Ekstraksi panjang *ciphertext* dari baris 1 elemen warna RGB piksel citra stego.

   Piksel (1, 1)

   216 = 1101 100**0**

   12 = 0000 110**0**

   46 = 0010 111**0**

   Piksel (1, 2)

   238= 1110 111**0**

   106= 0110 101**0**

   49 = 0011 000**1**

   Piksel (1, 3)

   60 = 0011 110**0**

   44 = 0010 110**0**

   Bit terekstrak = 0000 0100 = 4, berarti panjang *ciphertext* adalah 4 karakter. Hal ini berarti bahwa harus diekstrak 4 * 8 = 32 bit dari citra stego.

3. Ekstrak bit *ciphertext* mulai dari baris 2 elemen warna RGB piksel citra stego.

   Piksel (2, 1)

   94 = 0101 111**0**

   183 = 1011 011**1**

   178 = 1011 001**0**

   Piksel (2, 2)

   208 = 1101 000**0**

   136 = 1000 100**0**

   12 = 0000 110**0**

   Piksel (2, 3)

   141 = 1000 110**1**

   253 = 1111 110**1**

   82 = 0101 001**0**

Piksel (2, 4)

113 = 0111 000**1**

4 = 0000 010**0**

164 = 1010 010**0**

Piksel (2, 5)

14 = 0000 111**0**

126 = 0111 111**0**

153 = 1001 100**1**

Piksel (3, 1)

248 = 1111 100**0**

24 = 0001 100**0**

181 = 1011 010**1**

Piksel (3, 2)

246 = 1111 011**0**

233 = 1110 100**1**

108 = 0110 110**0**

Piksel (3, 3)

171 = 1010 101**1**

100 = 0110 010**0**

106 = 0110 101**0**

Piksel (3, 4)

190 = 1011 111**0**

73 = 0100 100**1**

242 = 1111 001**0**

Piksel (3, 5)

238 = 1110 111**0**

150 = 1001 011**0**

84 = 0101 010**0**

Piksel (4, 1)

151 = 1001 011**1**

192 = 1100 000**0**

Bit terekstrak: **0100 0011 0100 0010 0101 0100 0100 0010**

Ciphertext yang diperoleh:

0100 0011 = 67 = **C**

0100 0010 = 66 = **B**

0101 0100 = 84 = **T**

0100 0010 = 66 = **B**

Setelah diperoleh *ciphertext* terekstrak, maka proses akan diakhiri dengan mendekripsi *ciphertext* tersebut. Proses dekripsi dengan menggunakan metode *Playfair Cipher* dapat dirincikan sebagai berikut:

1. *Ciphertext* = **CBTB**
2. Kunci rahasia = **TES**
3. *Ciphertext* akan disusun menjadi bigram : **CB TB.**
4. Buat matriks kunci dengan ukuran 6 x 6, dengan cara urutkan sesuai dengan kemunculan huruf kunci, kemudian urutkan huruf belum terpakai pada kunci.

T	E	S	A	B	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 4.7 Matriks Kunci

5. Lakukan deskripsi dari cipherteks “CB” dan menghasilkan plainteks “BA”.

T	E	S	A	**B**	**C**
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

T	E	S	**A**	**B**	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 4.8 Matriks Deskripsi “CB”

6. Lakukan enkripsi dari cipherteks “TB” dan menghasilkan plainteks “CA”.

T	E	S	A	**B**	C
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

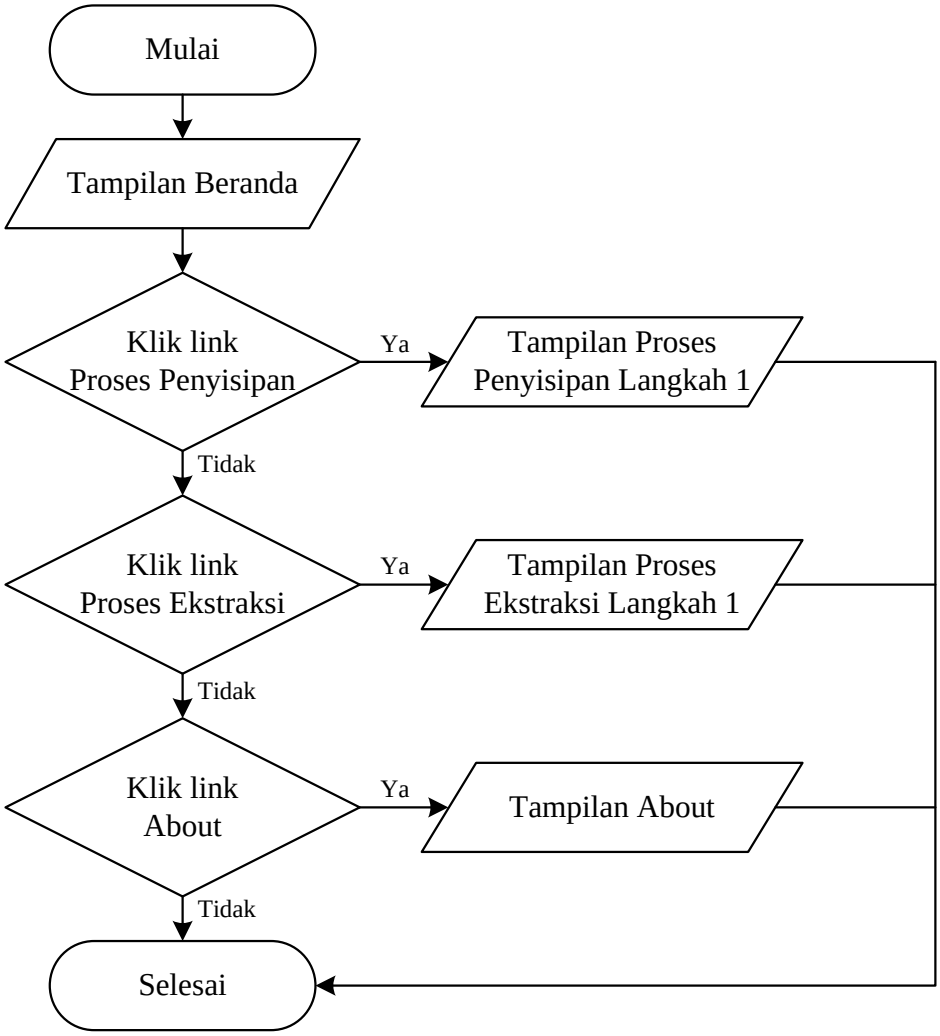
T	E	S	**A**	B	**C**
D	F	G	H	I	K
L	M	N	O	P	Q
R	U	V	W	X	Y
Z	␣	0	1	2	3
4	5	6	7	8	9

Gambar 4.9 Matriks Deskripsi “TB”

7. Plaintext yang diperoleh = **BACA**

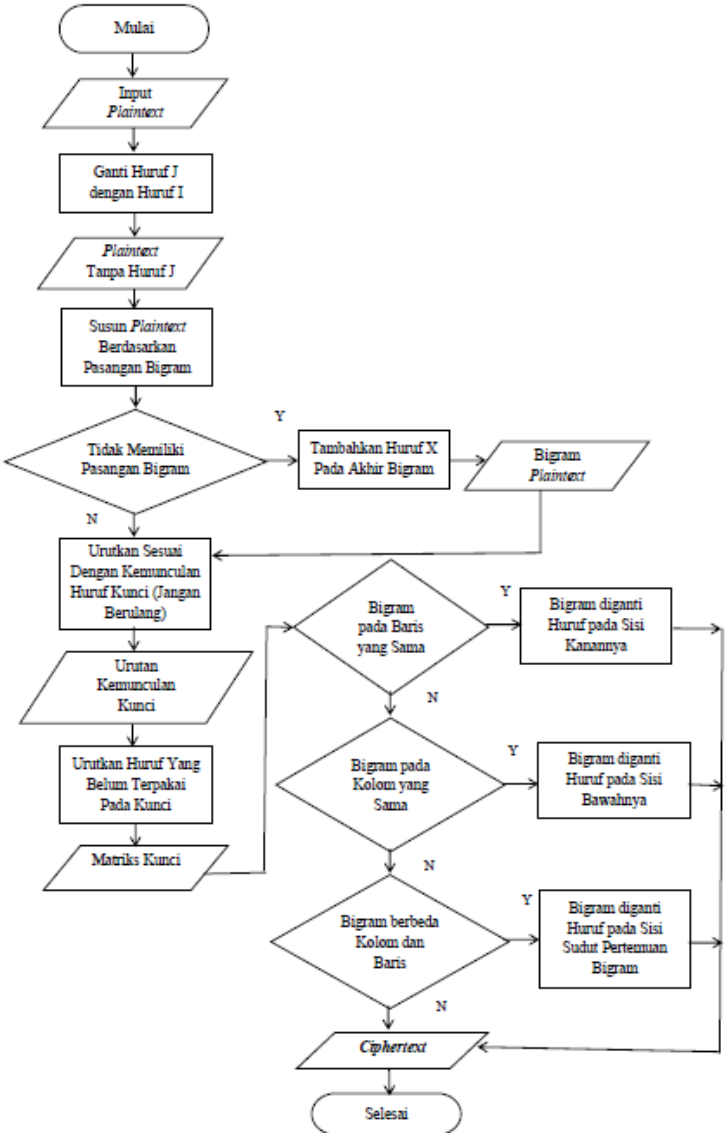
4.1.2. Flowchart

Proses kerja dari aplikasi dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



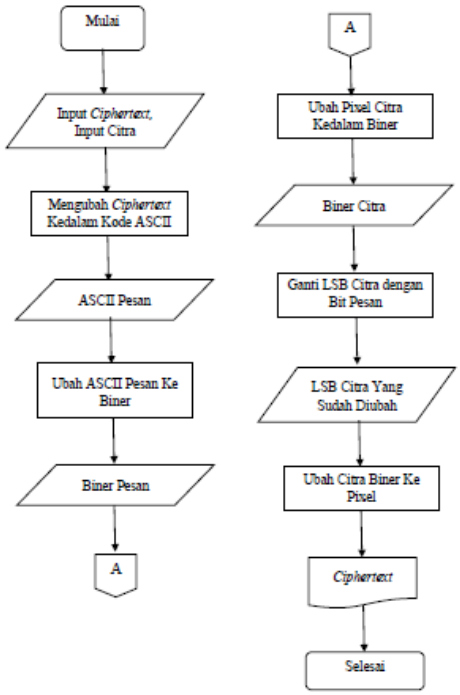
Gambar 4.10 Flowchart dari Tampilan Utama

Sementara itu, proses kerja dari prosedur enkripsi dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



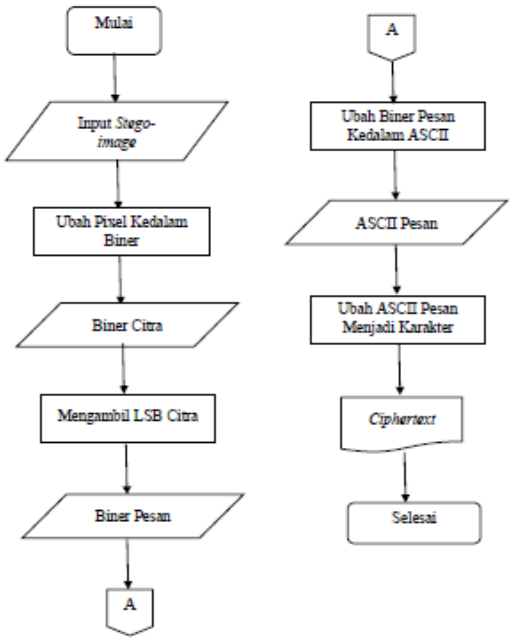
Gambar 4.11 Tampilan *Flowchart* dari Proses Enkripsi dengan Metode Playfair Cipher

Proses kerja dari proses penyisipan dengan metode LSB dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



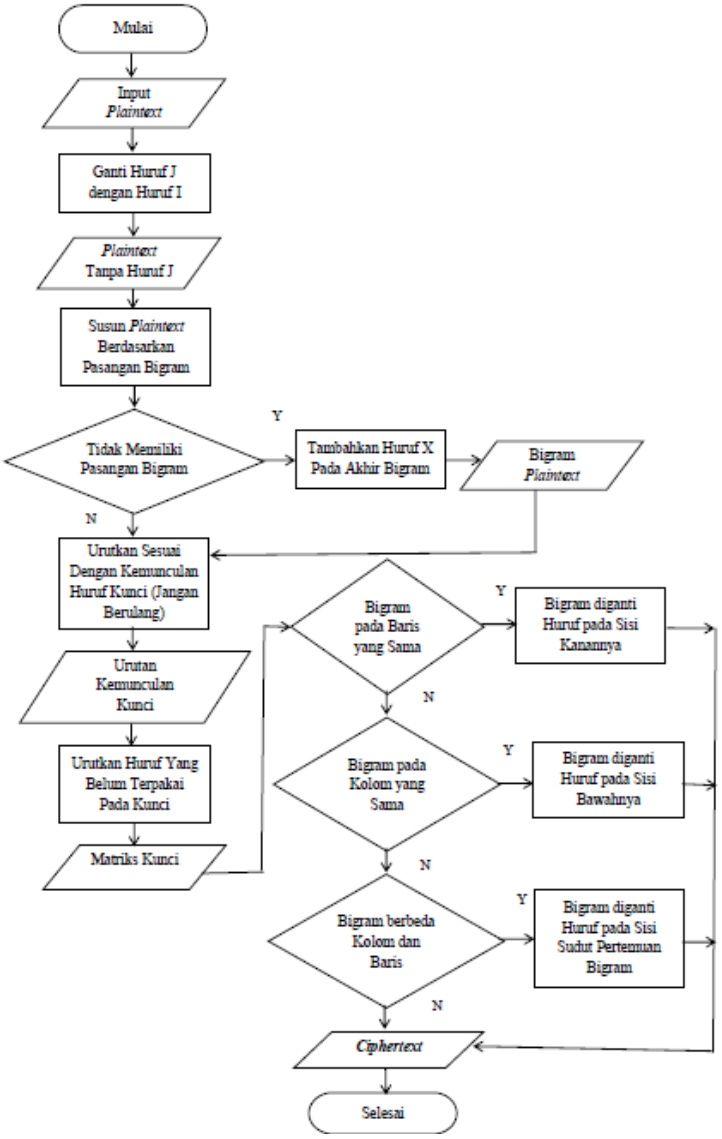
Gambar 4.12 Tampilan *Flowchart* dari Proses Penyisipan dengan Metode LSB

Proses kerja dari metode LSB dalam mengekstrak bit dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



Gambar 4.13 Tampilan *Flowchart* dari Proses Ekstraksi dengan Metode LSB

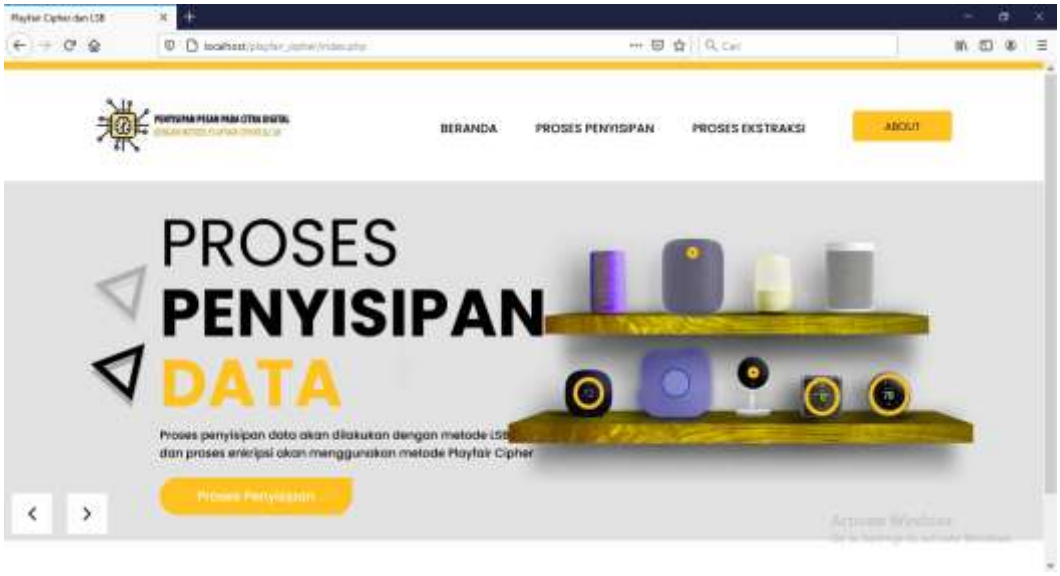
Proses kerja dari metode *Playfair Cipher* dalam mendekripsi *ciphertext* dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



Gambar 4.14 Tampilan *Flowchart* dari Proses Dekripsi dengan Metode Playfair Cipher

4.1.3. Tampilan Hasil Antarmuka Pemakai

Untuk menggunakan perangkat lunak ini, jalankan *browser* dengan mengakses alamat ”http://localhost/playfair_cipher/index.php”, maka akan ditampilkan tampilan utama dari program seperti terlihat pada gambar berikut:



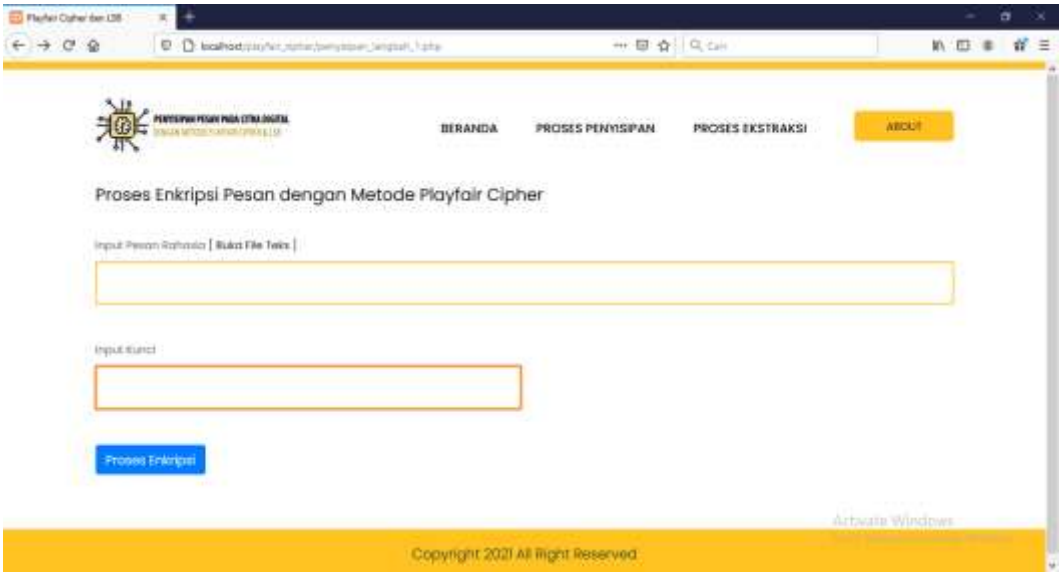
Gambar 4.15 Tampilan Utama

Pada tampilan utama ini terdapat beberapa menu yang berfungsi untuk mengakses halaman-halaman yang terdapat dalam sistem. Berikut perincian dari *link* yang terdapat dalam sistem:

1. Menu ‘Beranda’, yang berfungsi untuk menampilkan halaman Beranda.
2. Menu ‘Proses Penyisipan’, yang berfungsi untuk melakukan proses enkripsi dan penyisipan pesan ke dalam citra digital.
3. Menu ‘Proses Ekstraksi’, yang berfungsi untuk melakukan proses ekstraksi dan dekripsi *ciphertext*.
4. Menu ‘About’, yang berfungsi untuk menampilkan data dari pembuat aplikasi.

1. Proses Penyisipan Pesan

Untuk melakukan proses enkripsi dan penyisipan pesan, maka dapat mengklik menu ‘Proses Penyisipan’ sehingga sistem akan menampilkan halaman Penyisipan Langkah 1 seperti terlihat pada gambar 4.16.



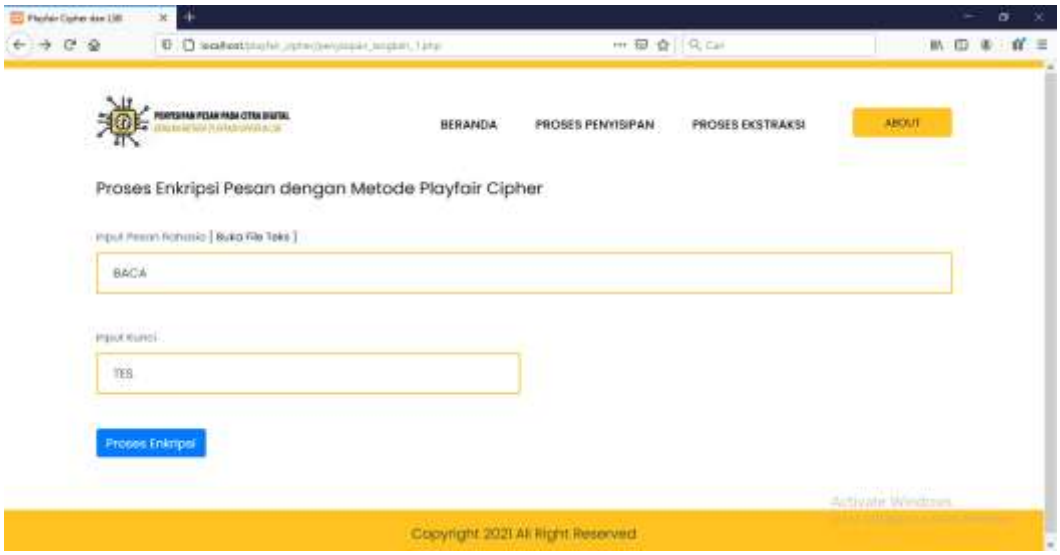
Gambar 4.16 Tampilan Proses Penyisipan Langkah 1

Pada halaman ini, *user* dapat memasukkan pesan rahasia yang akan dienkripsi dengan menggunakan metode *Playfair Cipher*. Untuk melakukan proses enkripsi juga diperlukan kunci rahasia yang akan digunakan untuk membentuk matriks berukuran 6 x 6. Setelah user memasukkan data pesan rahasia dan kunci, maka user dapat mengklik tombol Proses Enkripsi untuk memulai proses enkripsi.

Apabila *user* ingin memilih *file* teks, maka dapat mengklik *link* Buka *File* Teks, sehingga sistem akan menampilkan halaman *Browse File*, seperti terlihat pada gambar 4.17.

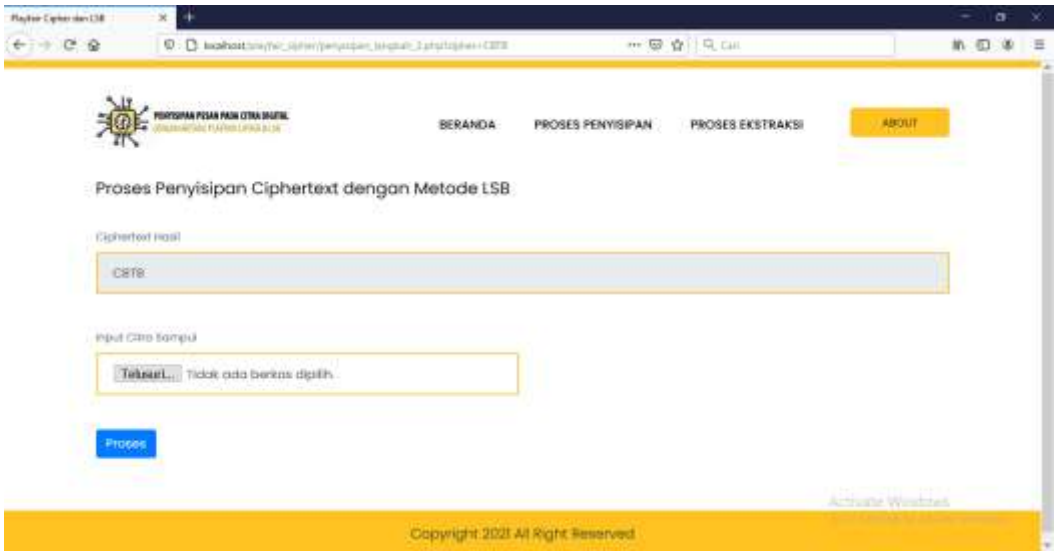
Gambar 4.17 Tampilan Halaman Browse File

Tampilan halaman Proses Penyisipan Langkah 1 setelah pengisian data dapat dilihat pada gambar 4.18.



Gambar 4.18 Tampilan Proses Penyisipan Langkah 1 Setelah Pengisian Data

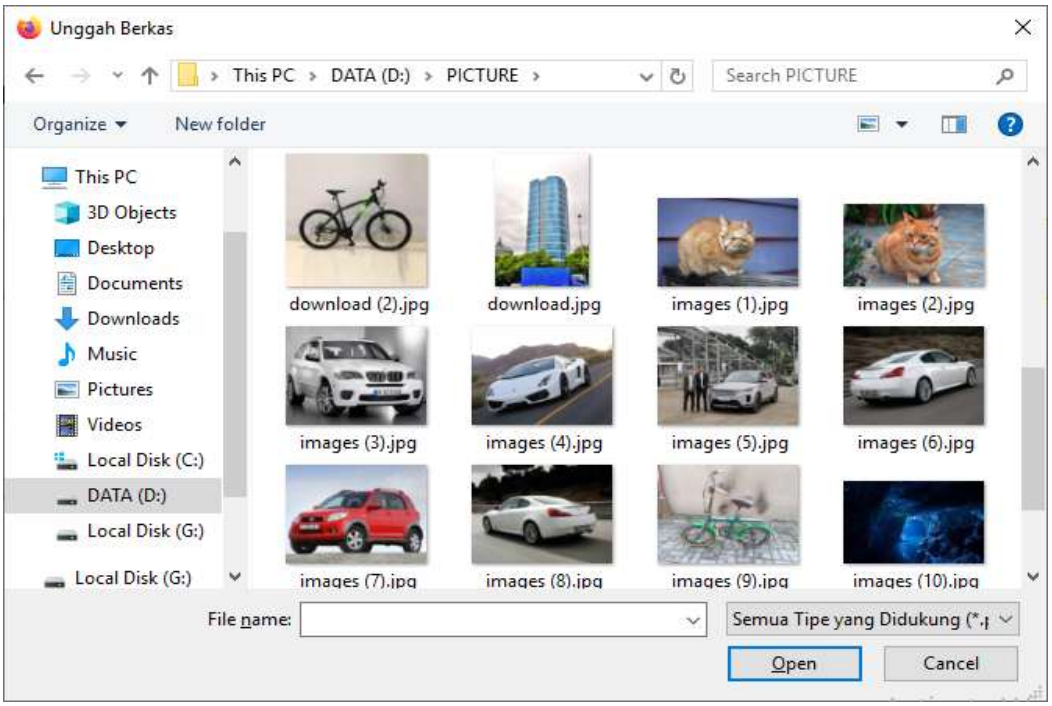
Setelah user mengklik tombol Proses Enkripsi, maka sistem akan menampilkan halaman Proses Penyisipan Halaman 2 seperti terlihat pada gambar 4.19.



Gambar 4.19 Tampilan Proses Penyisipan Langkah 2

Pada halaman Proses Penyisipan Langkah 2 ini, *user* dapat melihat *ciphertext* hasil enkripsi dengan menggunakan metode *Playfair Cipher*. Setelah

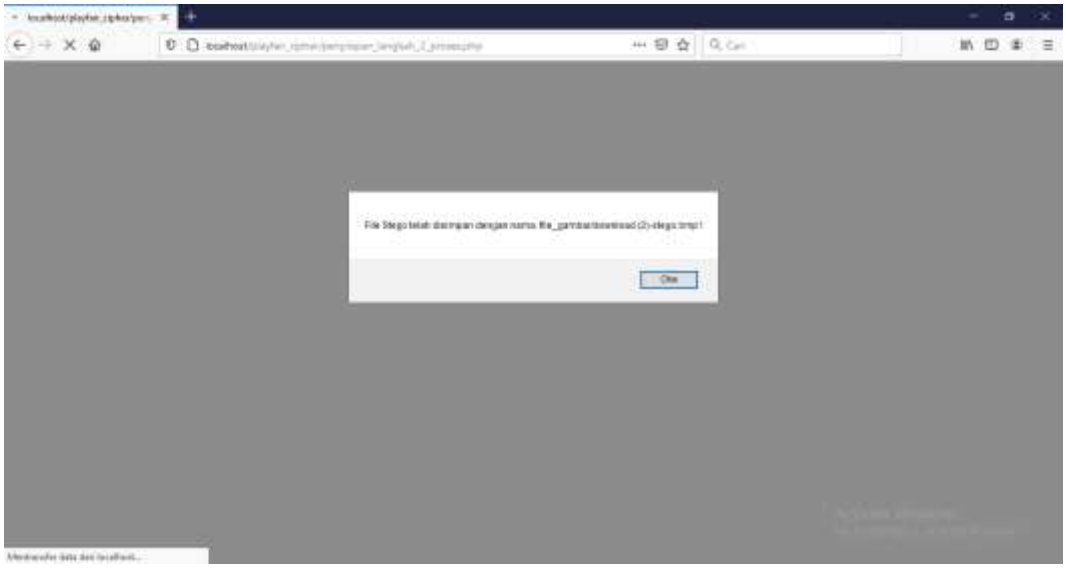
itu, *user* dapat memilih *file* citra sampul yang akan digunakan untuk menampung *ciphertext*. Caranya adalah dengan mengklik tombol Telusuri sehingga sistem akan menampilkan kotak dialog seperti terlihat pada gambar 4.20.



Gambar 4.20 Tampilan Kotak Dialog Pemilihan Citra Sampul

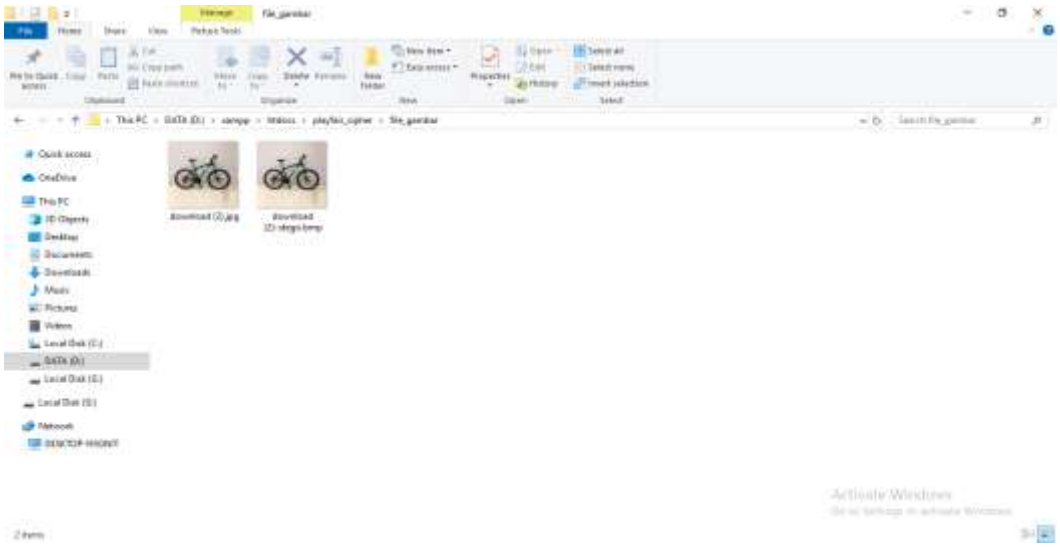
User dapat memilih *file* yang diinginkan dan klik tombol Open untuk membaca *file* citra tersebut. Setelah itu, user dapat mengklik tombol Proses yang terdapat pada halaman Proses Penyisipan Langkah 2 sehingga sistem akan menyisipkan *ciphertext* ke dalam citra sampul dengan menggunakan metode LSB.

Setelah proses penyisipan selesai, sistem akan menampilkan pesan pemberitahuan seperti terlihat pada gambar 4.21.



Gambar 4.21 Tampilan Pesan Pemberitahuan Bahwa Proses Penyisipan Selesai

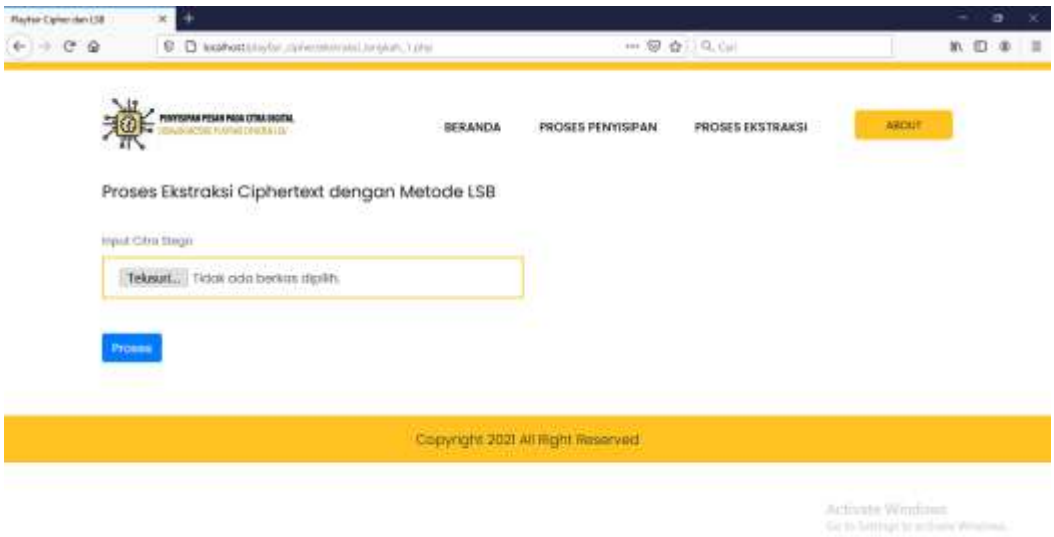
File citra stego yang dihasilkan beserta dengan *file* citra asli akan disimpan pada *folder* file_gambar, seperti terlihat pada gambar 4.22.



Gambar 4.22 Tampilan Windows Explorer yang Menampilkan Informasi File Citra Asli dan File Citra Stego

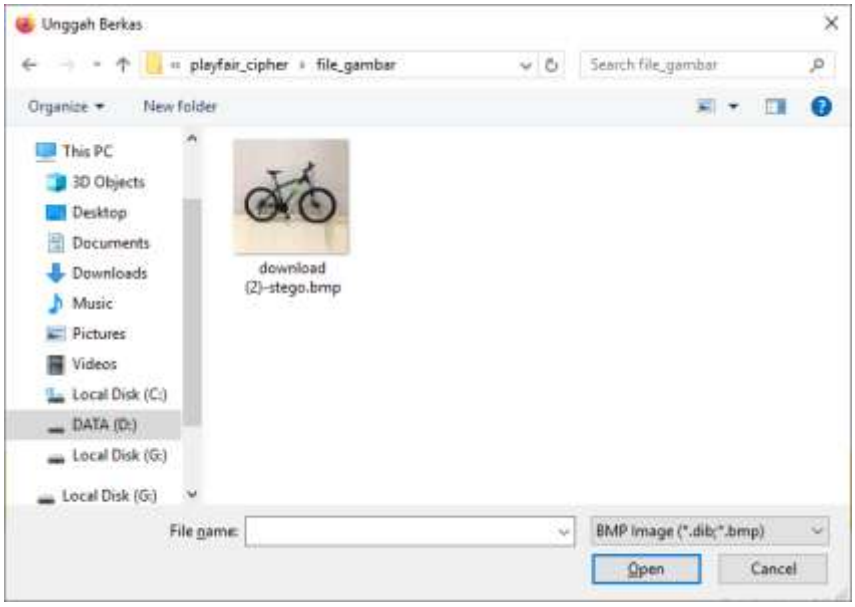
2. Proses Ekstraksi Pesan

Untuk melakukan proses ekstraksi pesan, maka dapat mengklik menu ‘Proses Ekstraksi’, sehingga sistem akan menampilkan halaman Ekstraksi Langkah 1 seperti terlihat pada gambar 4.23.



Gambar 4.23 Tampilan Proses Ekstraksi Langkah 1

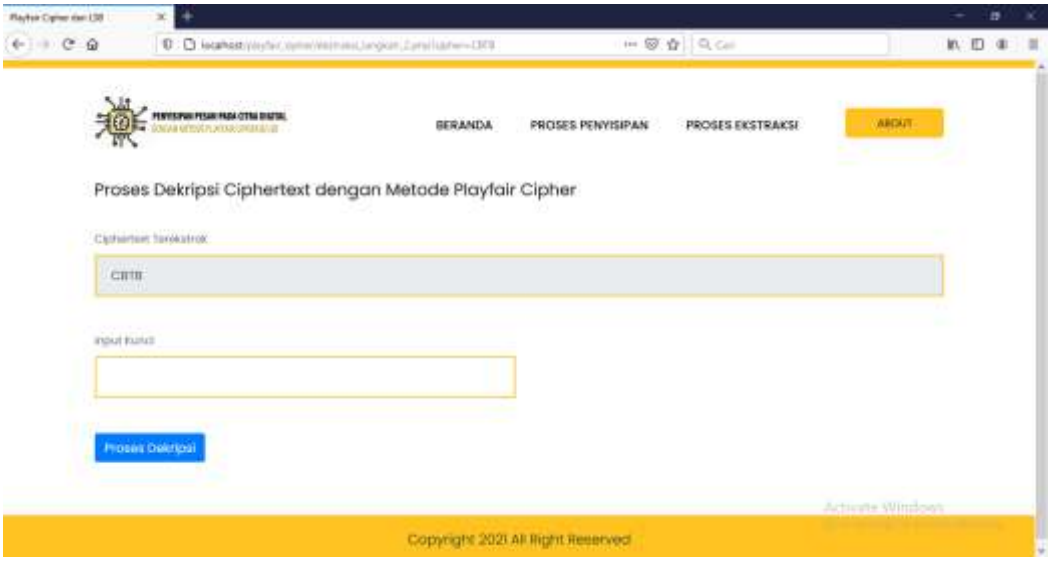
User dapat memilih *file* citra stego yang akan diekstrak *ciphertext*-nya dengan mengklik tombol Telusuri sehingga sistem akan menampilkan kotak dialog pemilihan citra stego seperti terlihat pada gambar 4.24.



Gambar 4.24 Tampilan Kotak Dialog Pemilihan Citra Stego

Setelah itu, *user* dapat mengklik tombol Open untuk membuka *file* citra stego yang dipilih. Kemudian, *user* dapat mengklik tombol Proses yang terdapat pada halaman Ekstraksi Langkah 1, sehingga sistem akan mengekstrak ciphertext

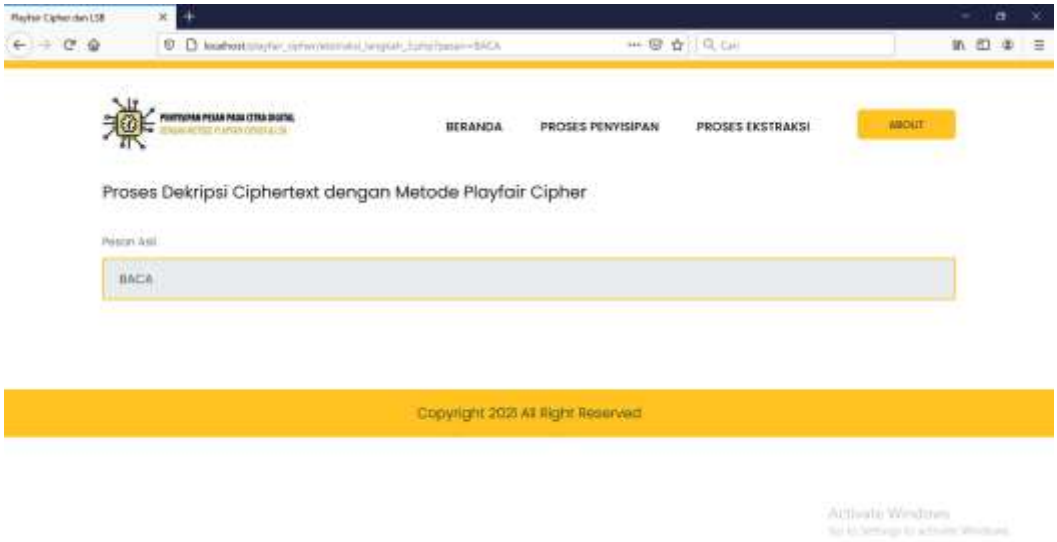
keluar dari citra stego dan menampilkannya pada halaman Ekstraksi Langkah 2 seperti terlihat pada gambar 4.25.



Gambar 4.25 Tampilan Halaman Ekstraksi Langkah 2

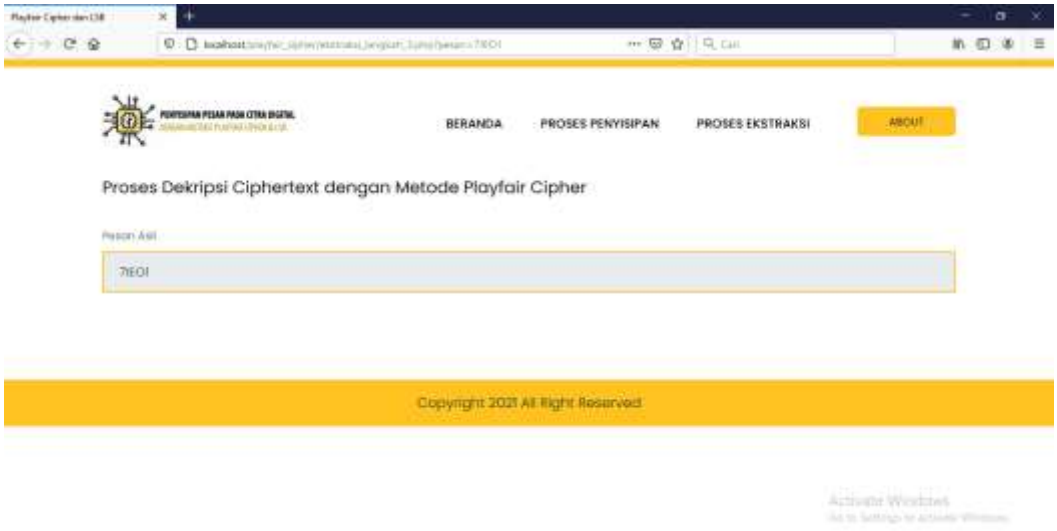
Pada halaman Ekstraksi Langkah 2 ini, *user* dapat memasukkan kunci yang akan digunakan untuk mendekripsi *ciphertext*. Kunci yang dimasukkan harus sama seperti kunci yang digunakan pada proses enkripsi. Apabila kunci yang dimasukkan tidak sama, maka pesan asli tidak dapat diperoleh kembali. Setelah itu, user dapat mengklik tombol Proses Dekripsi untuk melakukan proses dekripsi *ciphertext*.

Tampilan halaman Ekstraksi Langkah 3 saat pengisian kunci rahasia yang benar dapat dilihat pada gambar 4.26.



Gambar 4.26. Tampilan Halaman Ekstraksi Langkah 3 dengan Pengisian Kunci yang Benar

Sementara itu, tampilan halaman ekstraksi langkah 3 pada saat pengisian kunci yang salah dapat dilihat pada gambar 4.27.



Gambar 4.27 Tampilan Halaman Ekstraksi Langkah 3 dengan Pengisian Kunci yang Salah

Setelah dilakukanya perencanaan dan penerapan penggunaan aplikasi ini dapat dikatakan berhasil. Karena aplikasi ini dapat mengamankan pesan rahasia dengan cara mengacak isi pesan tersebut menggunakan metode *playfair cipher*

sehingga pesan tersebut sulit di baca dan menyembunyikan pesan rahasia kedalam sebuah citra menggunakan metode *least significant bit* sehingga tidak terlihat secara kasat mata.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah menyelesaikan pembuatan perangkat lunak ini, penulis dapat menarik beberapa kesimpulan sebagai berikut:

1. Aplikasi dapat digunakan untuk menyembunyikan pesan rahasia pada citra digital.
2. Panjangnya pesan yang dapat disisipkan tergantung pada ukuran citra digital yang digunakan.
3. Perbedaan warna citra input dan citra hasil juga tidak kelihatan jelas.
4. Hasil *output* citra digital hanya berupa citra berformat BMP, karena proses penyimpanan data ke bentuk JPG akan mengubah warna piksel citra digital sehingga informasi yang disisipkan menjadi rusak atau hilang.
5. Pesan asli yang didapat dari proses deskripsi akan berubah menjadi huruf kapital hal ini terjadi akibat matriks kunci yang digunakan adalah huruf kapital.

5.2. Saran

Penulis ingin memberikan beberapa saran yang mungkin berguna untuk pengembangan lebih lanjut pada perangkat lunak, yaitu :

1. Perangkat lunak dapat dikembangkan lagi dengan menambahkan fitur lainnya seperti fitur tutorial yang mampu menjelaskan prosedur kerja dari algoritma yang dibahas secara terperinci.
2. Perangkat lunak dapat dikembangkan dengan membandingkan algoritma steganografi yang dibahas dengan algoritma lainnya yang sejenis untuk mengetahui kelebihan dan kelemahan algoritma yang dibuat.

DAFTAR PUSTAKA

Furqan, Mhd., Sriani,. Sari, Indah Eka Yulia,. *Penerapan Metode Otsu dalam Melakukan Segmentasi Citra Pada Citra Naskah Arab*. Dalam jurnal *Matrik: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer* ISSN: 2476-9843 *Vol.20, No.1*.

Hafiz, Aliy. 2019. *Steganografi Berbasis Citra Digital Untuk Menyembunyikan Data Menggunakan Metode Least Significant Bit (LSB),* dalam jurnal *Jurnal Cendikia Vol. XVII.*

Hermawati, Fajar Astuti. 2013 .*Pengolahan Citra Digital Konsep & Teori*. CV. Andi Offset :Yogyakarta.

Indrajit, Richardus Eka. 2014. *Konsep dan Strategi Keamanan Informasi Di Dunia Cyber*. Graha Ilmu : Yogyakarta.

Laoli Desimer, dkk. 2020. *Penerapan Algoritma Hill Cipher Dan Least Significant Bit (LSB) Untuk Pengamanan Pesan Pada Citra Digital. Jurnal JISKa Vol. 4, No.3* . Medan: STMIK Pelita Nusantara.

Muljoto, dkk. 2017. *Pengolahan Citra Digital*. CV.Andi Offset : Yogyakarta.

Munir, R. 2006. *Kriptografi, Cetakan Pertama*. Penerbit Informatika : Bandung.

Murdowo, Sugeng. 2020 . *Manual Perhitungan Menggunakan Kriptografi Klasik Playfair Cipher*. Semarang .*Jurnal INFOKAM Vol. XVI, No.1*.

Nasution, Yusuf Ramadhan,. Furqan, Mhd,. Sinaga, Meri. *Metode Spread Spectrum Dalam Pengamanan Data Teks Pada Citra Digital*. Dalam jurnal *Jurnal Sains Komputer & Informatika (J-Sakti)* ISSN: 2548-9771 *Volume. 4, Nomer. 2.*

Sadikin, Rifki. 2012. *Kriptografi Untuk Keamanan Jaringan*. CV Andi Offset : Yogyakarta.

Setyaningsih, E. 2015. *Kriptografi & Implementasi Menggunakan Matlab*. CV Andi Offset: Yogyakarta.

Stalling, W.2010. *Cryptography and Network Security: Principles and Practice*. 5thedition: Practice Hall.

Simbolon, Ratna Wati. 2016. *Pengaman Transkrip Nilai Mahasiswa Menggunakan Kriptografi Playfair Cipher dan Steganografi Dengan Teknik Least Significant Bit (LSB). Jurnal Teknologi Informasi Dan Komunikasi Vol. 5, No.1.*

Sitorus, Michel . 2015. *Teknik Steganografi dengan Metode Least Significant Bit (LSB), Jurnal Ilmiah Fakultas Teknik LIMIT'S ISSN 0612- 1184* Vol.11, No.2.

Sumarno, dkk. 2018. *Pengamanan Berkas Dokumen Menggunakan Fungsi Algoritma Steganografi LSB. Jurnal Ilmu Komputer dan Informatika Vol. 2, No. 01.* Pematang Siantar: AMIK Tunas Bangsa.

Wiyata. 2016. *Implementasi Steganografi Metode LSB Menggunakan Program PHP Untuk Keamanan Pesan Gambar*. *Jurnal ICT Learning Vol.2, No.2.* Jakarta Selatan: Universitas Budi Luhur.

LAMPIRAN

ListingProgram

1. PROSESENKRIPSI

```
<?php

$pesan=trim($_POST["pesan"]);
$kunci=trim($_POST["kunci"]);

if(strlen($pesan)==0)
{
     echo"
     <scripttype='text/javascript'>
          alert('Inputpesanmasihkosong!');
          window.location='penyisipan_langkah_1.php';
     </script>
     ";
}

//Ubahmenjadihurufkapitalsemua
$pesan=strtoupper($pesan);

//HurufJgantijadihurufI
$pesan=str_replace("J","I",$pesan);

//Plainteksdisusunmenjadibigram
$bigram=array();$idx=0;
for($i=0;$i<strlen($pesan);$i+=2)
{
     if($i+2<=strlen($pesan))
          $bigram[$idx]=substr($pesan,$i,2);
     else
          $bigram[$idx] = substr($pesan, $i, 1) . "X";
//TambahkankarakterXkalaucuma1karakterdalambigram

     $idx+=1;
}

//Buatmatrikskunci6x6
$string_elemen="ABCDEFGHIKLMNOPQRSTUVWXYZ0123456789";
$string_elemen_hasil=$kunci;

//Susunurutandaristringpadamatriks
for($i=0;$i<strlen($string_elemen);$i++)
{
     $kar=substr($string_elemen,$i,1);
     $pos=strpos($string_elemen_hasil,$kar);
     if($pos===false)
     {
          //Karaktertidakditemukan->masukkankestringhasil
```

```
            $string_elemen_hasil.=$kar;
      }
}

//Bentukstringjadimatriks6x6
$matriks_kunci=array();$idx=0;
for($i=0;$i<6;$i++)
      for($j=0;$j<6;$j++)
      {
            $matriks_kunci[$i][$j]                                    =
substr($string_elemen_hasil,$idx,1);
            $idx+=1;
      }

//Fungsi pengambilan baris, kolom dari karakter pada matriks
kunci
functionGetRC($karakter,$arr_matriks_kunci)
{
      for($i=0;$i<6;$i++)
            for($j=0;$j<6;$j++)
            {
                  if($arr_matriks_kunci[$i][$j]==$karakter)
                        return$i.",".$j;
            }

      return"";
}

//LakukanprosesenkripsidenganPlayfairCipher
$bigram_hasil=array();$ciphertext="";
for($i=0;$i<sizeof($bigram);$i++)
{
      $kar1=substr($bigram[$i],0,1);
      $kar2=substr($bigram[$i],1,1);

      $RC1=GetRC($kar1,$matriks_kunci);
      $RC2=GetRC($kar2,$matriks_kunci);

      $temp=explode(",",$RC1);
      $brs1=$temp[0];
      $klm1=$temp[1];

      $temp=explode(",",$RC2);
      $brs2=$temp[0];
      $klm2=$temp[1];

      if($brs1==$brs2)
      {
            //kedua karakter berada pada baris yang sama --> geser
ke kanan
            $klm1_hasil=($klm1+1)%6;
```

```
            $klm2_hasil=($klm2+1)%6;
            $brs1_hasil=$brs1;
            $brs2_hasil=$brs2;

      }
      elseif($klm1==$klm2)
      {
            //kedua karakter berada pada kolom yang sama -->geser
kebawah
            $brs1_hasil=($brs1+1)%6;
            $brs2_hasil=($brs2+1)%6;
            $klm1_hasil=$klm1;
            $klm2_hasil=$klm2;
      }
      else
      {
            //karaktermembentukkotak
            $brs1_hasil=$brs1;
            $brs2_hasil=$brs2;
            $klm1_hasil=$klm2;
            $klm2_hasil=$klm1;
      }

      $bigram_hasil[$i]                                          =
$matriks_kunci[$brs1_hasil][$klm1_hasil]                         .
$matriks_kunci[$brs2_hasil][$klm2_hasil];



      //Tambahkankehasilciphertext
      $ciphertext.=$bigram_hasil[$i];
}

      echo"
      <scripttype='text/javascript'>

      window.location='penyisipan_langkah_2.php?cipher=$ciph
ertext';
      </script>
      ";
?>
```

2. PENYISIPAN PESAN KEDALAM GAMBAR

```
<?php

$ciphertext=$_POST["ciphertext"];

$target_dir="file_gambar/";
```

```
$target_file = $target_dir .
basename($_FILES["gambar"]["name"]);
$uploadOk=1;
$imageFileType =
strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

//Checkifimagefileisaactualimageorfakeimage
if(isset($_POST["submit"])){
$check=getimagesize($_FILES["gambar"]["tmp_name"]);
if($check!==false){
echo"Fileisanimage-".$check["mime"].".";
$uploadOk=1;
}else{
echo"Fileisnotanimage.";
$uploadOk=0;
}
}

//Checkiffilealreadyexists
if(file_exists($target_file)){
echo"Sorry,filealreadyexists.";
$uploadOk=0;
}

//Checkfilesize
if($_FILES["gambar"]["size"]>5000000){
echo"Sorry,yourfileistoolarge.";
$uploadOk=0;
}

//Allowcertainfileformats
if($imageFileType != "jpg" && $imageFileType != "png" &&
$imageFileType!="jpeg"
&&$imageFileType!="gif"){
echo"Sorry,onlyJPG,JPEG,PNG&GIFfilesareallowed.";
$uploadOk=0;
}

//Checkif$uploadOkissetto0byanerror
if($uploadOk==0){
echo"Sorry,yourfilewasnotuploaded.";
//ifeverythingisok,trytouploadfile
}else{
    if (move_uploaded_file($_FILES["gambar"]["tmp_name"],
$target_file)){
        //echo "The file ". htmlspecialchars(basename(
$_FILES["gambar"]["name"])). "hasbeenuploaded.";
}else{
echo"Sorry,therewasanerroruploadingyourfile.";
}
}
```

```
//Sisipkan ciphertext ke gambar
//Ubah ciphertext ke deretan biner
$biner="";$bin="";
for($i=0;$i<strlen($ciphertext);$i++)
{
     $kar=substr($ciphertext,$i,1);
     $asc=ord($kar);
     $bin=decbin($asc);
     if(strlen($bin)<8)
          $bin=str_repeat("0",8-strlen($bin)).$bin;
     $biner.=$bin;
}

//Baca warna piksel citra
$resource=file_get_contents($target_file);
$imgs=getimagesizefromstring($resource);
$image_width=$imgs[0];
$image_height=$imgs[1];

$ext=$imageFileType;
$lokasi=$target_file;
if((strtolower($ext)=="jpg")||(strtolower($ext)=="jpeg"))
     $imgs=imagecreatefromjpeg($lokasi);
elseif(strtolower($ext)=="png")
     $imgs=imagecreatefrompng($lokasi);
elseif(strtolower($ext)=="bmp")
     $imgs=imagecreatefrombmp($lokasi);
elseif(strtolower($ext)=="gif")
     $imgs=imagecreatefromgif($lokasi);

//Panjang ciphertext
$cipher_length=decbin(strlen($ciphertext));
if(strlen($cipher_length)<8)$cipher_length=str_repeat("0",
8-strlen($cipher_length)).$cipher_length;

$originalArray=array();
//Baca warna piksel citra
for($y=0;$y<$image_height;$y++)
     for($x=0;$x<$image_width;$x++)
     {
          //pixel color at (x,y)
          $color=imagecolorat($imgs,$x,$y);

          $r=($color>>16)&0xFF;
          $g=($color>>8)&0xFF;
          $b=$color&0xFF;

          $originalArray[$y][$x][0]=$r;
          $originalArray[$y][$x][1]=$g;
```

```
            $originalArray[$y][$x][2]=$b;
        }

//pixelcolor1
$r1=decbin($originalArray[0][0][0]);
$g1=decbin($originalArray[0][0][1]);
$b1=decbin($originalArray[0][0][2]);

if(strlen($r1)<8)$r1=str_repeat("0",8-strlen($r1)).$r1;
if(strlen($g1)<8)$g1=str_repeat("0",8-strlen($g1)).$g1;
if(strlen($b1)<8)$b1=str_repeat("0",8-strlen($b1)).$b1;

//Sisipkankegambar
$originalArray[0][0][0]   =   bindec(substr($r1,   0,   7)   .
substr($cipher_length,0,1));
$originalArray[0][0][1]   =   bindec(substr($g1,   0,   7)   .
substr($cipher_length,1,1));
$originalArray[0][0][2]   =   bindec(substr($b1,   0,   7)   .
substr($cipher_length,2,1));

//pixelcolor2
$r1=decbin($originalArray[0][1][0]);
$g1=decbin($originalArray[0][1][1]);
$b1=decbin($originalArray[0][1][2]);

if(strlen($r1)<8)$r1=str_repeat("0",8-strlen($r1)).$r1;
if(strlen($g1)<8)$g1=str_repeat("0",8-strlen($g1)).$g1;
if(strlen($b1)<8)$b1=str_repeat("0",8-strlen($b1)).$b1;

//Sisipkankegambar
$originalArray[0][1][0]   =   bindec(substr($r1,   0,   7)   .
substr($cipher_length,3,1));
$originalArray[0][1][1]   =   bindec(substr($g1,   0,   7)   .
substr($cipher_length,4,1));
$originalArray[0][1][2]   =   bindec(substr($b1,   0,   7)   .
substr($cipher_length,5,1));

//pixelcolor3
$r1=decbin($originalArray[0][2][0]);
$g1=decbin($originalArray[0][2][1]);
$b1=decbin($originalArray[0][2][2]);

if(strlen($r1)<8)$r1=str_repeat("0",8-strlen($r1)).$r1;
if(strlen($g1)<8)$g1=str_repeat("0",8-strlen($g1)).$g1;
if(strlen($b1)<8)$b1=str_repeat("0",8-strlen($b1)).$b1;

//Sisipkankegambar
$originalArray[0][2][0]   =   bindec(substr($r1,   0,   7)   .
substr($cipher_length,6,1));
$originalArray[0][2][1]   =   bindec(substr($g1,   0,   7)   .
substr($cipher_length,7,1));
```

```
//Sisipkan ciphertext ke gambar
$idx=0;
for ($y=1;$y<$image_height;$y++)
     for ($x=0;$x<$image_width;$x++)
     {

          $r1=decbin($originalArray[$y][$x][0]);
          $g1=decbin($originalArray[$y][$x][1]);
          $b1=decbin($originalArray[$y][$x][2]);

          if (strlen($r1) < 8) $r1 = str_repeat("0", 8 -
strlen($r1)).$r1;
          if (strlen($g1) < 8) $g1 = str_repeat("0", 8 -
strlen($g1)).$g1;
          if (strlen($b1) < 8) $b1 = str_repeat("0", 8 -
strlen($b1)).$b1;

          if($idx<strlen($biner))
          {
               //Sisipkan ke gambar elemen R
               $originalArray[$y][$x][0]                    =
bindec(substr($r1,0,7).substr($biner,$idx,1));
               $idx+=1;
          }

          if($idx<strlen($biner))
          {
               //Sisipkan ke gambar elemen G
               $originalArray[$y][$x][1]                    =
bindec(substr($g1,0,7).substr($biner,$idx,1));
               $idx+=1;
          }

          if($idx<strlen($biner))
          {
               //Sisipkan ke gambar elemen B
               $originalArray[$y][$x][2]                    =
bindec(substr($b1,0,7).substr($biner,$idx,1));
               $idx+=1;
          }

     }

//Set warna ke citra hasil
$output_image      =      imagecreatetruecolor($image_width,
$image_height);
for ($x=0;$x<$image_width;$x++) {
 for ($y=0;$y<$image_height;$y++) {
```

```
                $orgb = imagecolorallocate($output_image,
$originalArray[$y][$x][0],       $originalArray[$y][$x][1],
$originalArray[$y][$x][2]);
 imagesetpixel($output_image,$x,$y,$orgb);
 }
}

//Savetheimage
$path_parts=pathinfo($target_file);
$file_hasil = $target_dir . $path_parts['filename'] . "-
stego.bmp";
imagebmp($output_image,$file_hasil,false);

     echo"
     <scripttype='text/javascript'>
          alert('File Stego telah disimpan dengan nama:
$file_hasil!');
          window.location='index.php';
     </script>
     ";
?>
```

3. PROSES EKSTRAKSI PESAN

```
<?php

$target_dir="file_gambar/";
$target_file          =          $target_dir          .
basename($_FILES["gambar"]["name"]);

//Bacawarnapikselcitra
$resource=file_get_contents($target_file);
$imgs=getimagesizefromstring($resource);
$image_width=$imgs[0];
$image_height=$imgs[1];

$imageFileType                                          =
strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
$ext=$imageFileType;
$lokasi=$target_file;
if(strtolower($ext)=="bmp")
     $imgs=imagecreatefrombmp($lokasi);
else
{
     echo"
     <scripttype='text/javascript'>
          alert('FileStegoharusdalamformatbmp!');
          window.location='ekstraksi_langkah_1.php';
     </script>
     ";
```

```
}

$originalArray=array();
if(strtolower($ext)=="bmp")
{
     //Bacawarnapikselcitra
     for($y=0;$y<$image_height;$y++)
          for($x=0;$x<$image_width;$x++)
          {
               //pixelcolorat(x,y)
               $color=imagecolorat($imgs,$x,$y);

               $r=($color>>16)&0xFF;
               $g=($color>>8)&0xFF;
               $b=$color&0xFF;

               $originalArray[$y][$x][0]=$r;
               $originalArray[$y][$x][1]=$g;
               $originalArray[$y][$x][2]=$b;
          }

     //Ekstrakpanjangciphertext
     $cipher_length="";

     $r1=decbin($originalArray[0][0][0]);
     if(strlen($r1)<8)$r1=str_repeat("0",8-strlen($r1)).
$r1;

     $g1=decbin($originalArray[0][0][1]);
     if(strlen($g1)<8)$g1=str_repeat("0",8-strlen($g1)).
$g1;

     $b1=decbin($originalArray[0][0][2]);
     if(strlen($b1)<8)$b1=str_repeat("0",8-strlen($b1)).
$b1;

     $cipher_length .= substr($r1,7, 1) . substr($g1,7, 1) .
substr($b1,7,1);


     $r1=decbin($originalArray[0][1][0]);
     if(strlen($r1)<8)$r1=str_repeat("0",8-strlen($r1)).
$r1;

     $g1=decbin($originalArray[0][1][1]);
     if(strlen($g1)<8)$g1=str_repeat("0",8-strlen($g1)).
$g1;

     $b1=decbin($originalArray[0][1][2]);
     if(strlen($b1)<8)$b1=str_repeat("0",8-strlen($b1)).
$b1;
```

```
    $cipher_length .= substr($r1,7, 1) . substr($g1,7, 1) .
substr($b1,7,1);

    $r1=decbin($originalArray[0][2][0]);
    if (strlen($r1) < 8) $r1 = str_repeat("0", 8 - strlen($r1)) .
$r1;

    $g1=decbin($originalArray[0][2][1]);
    if (strlen($g1) < 8) $g1 = str_repeat("0", 8 - strlen($g1)) .
$g1;

    $cipher_length.=substr($r1,7,1).substr($g1,7,1);
    $panjang_ciphertext=bindec($cipher_length)*8;


    //Ekstraksibitciphertext
    $piksel_biner="";
    for($y=1;$y<$image_height;$y++)
          for($x=0;$x<$image_width;$x++)
          {
                $r1=decbin($originalArray[$y][$x][0]);
                if (strlen($r1) < 8) $r1 = str_repeat("0", 8 -
strlen($r1)).$r1;

                $g1=decbin($originalArray[$y][$x][1]);
                if (strlen($g1) < 8) $g1 = str_repeat("0", 8 -
strlen($g1)).$g1;

                $b1=decbin($originalArray[$y][$x][2]);
                if (strlen($b1) < 8) $b1 = str_repeat("0", 8 -
strlen($b1)).$b1;

                if          (strlen($piksel_biner)          <
$panjang_ciphertext)$piksel_biner.=substr($r1,7,1);
                if          (strlen($piksel_biner)          <
$panjang_ciphertext)$piksel_biner.=substr($g1,7,1);
                if          (strlen($piksel_biner)          <
$panjang_ciphertext)$piksel_biner.=substr($b1,7,1);

          }

    //Kelompokkanmenjadisubblok8bit
    $bit_ekstrak="";$ciphertext="";
    for($i=0;$i<strlen($piksel_biner);$i+=8)
    {
          $bit_ekstrak=substr($piksel_binPer,$i,8);
          $ciphertext.=chr(bindec($bit_ekstrak));
    }

//    echo$piksel_biner;
```

```
            echo"
            <scripttype='text/javascript'>

      window.location='ekstraksi_langkah_2.php?cipher=$ciphe
rtext';
            </script>
            ";
}
?>
```

4. PROSESDESKRIPSI

```
<?php

$ciphertext=trim($_POST["ciphertext"]);
$kunci=trim($_POST["kunci"]);

//Ciphertextdisusunmenjadibigram
$bigram=array();$idx=0;
for($i=0;$i<strlen($ciphertext);$i+=2)
{
      $bigram[$idx]=substr($ciphertext,$i,2);
      $idx+=1;
}

//Buatmatrikskunci6x6
$string_elemen="ABCDEFGHIKLMNOPQRSTUVWXYZ0123456789";
$string_elemen_hasil=$kunci;

//Susunurutandaristringpadamatriks
for($i=0;$i<strlen($string_elemen);$i++)
{
      $kar=substr($string_elemen,$i,1);
      $pos=strpos($string_elemen_hasil,$kar);
      if($pos===false)
      {
            //Karaktertidakditemukan->masukkankestringhasil
            $string_elemen_hasil.=$kar;
      }
}

//Bentukstringjadimatriks6x6
$matriks_kunci=array();$idx=0;
for($i=0;$i<6;$i++)
      for($j=0;$j<6;$j++)
      {
            $matriks_kunci[$i][$j]                              =
substr($string_elemen_hasil,$idx,1);
```

```
            $idx+=1;
      }

//Fungsi pengambilan baris, kolom dari karakter pada matriks
kunci
functionGetRC($karakter,$arr_matriks_kunci)
{
      for($i=0;$i<6;$i++)
            for($j=0;$j<6;$j++)
            {
                  if($arr_matriks_kunci[$i][$j]==$karakter)
                        return$i.",".$j;
            }

      return"";
}

//Lakukanprosesenkripsidengan PlayfairCipher
$bigram_hasil=array();$pesan="";
for($i=0;$i<sizeof($bigram);$i++)
{
      $kar1=substr($bigram[$i],0,1);
      $kar2=substr($bigram[$i],1,1);

      $RC1=GetRC($kar1,$matriks_kunci);
      $RC2=GetRC($kar2,$matriks_kunci);

      $temp=explode(",",$RC1);
      $brs1=$temp[0];
      $klm1=$temp[1];

      $temp=explode(",",$RC2);
      $brs2=$temp[0];
      $klm2=$temp[1];

      if($brs1==$brs2)
      {
            //kedua karakter berada pada baris yang sama -->geser
kekiri
            $klm1_hasil=($klm1-1);
            if($klm1_hasil<0)$klm1_hasil+=6;
            $klm2_hasil=($klm2-1);
            if($klm2_hasil<0)$klm2_hasil+=6;
            $brs1_hasil=$brs1;
            $brs2_hasil=$brs2;

      }
      elseif($klm1==$klm2)
      {
            //kedua karakter berada pada kolom yang sama -->geser
keatas
```

```
            $brs1_hasil=($brs1-1);
            if($brs1_hasil<0)$brs1_hasil+=6;
            $brs2_hasil=($brs2-1);
            if($brs2_hasil<0)$brs2_hasil+=6;
            $klm1_hasil=$klm1;
            $klm2_hasil=$klm2;
      }
      else
      {
            //karaktermembentukkotak
            $brs1_hasil=$brs1;
            $brs2_hasil=$brs2;
            $klm1_hasil=$klm2;
            $klm2_hasil=$klm1;
      }

      $bigram_hasil[$i]                                          =
$matriks_kunci[$brs1_hasil][$klm1_hasil]                         .
$matriks_kunci[$brs2_hasil][$klm2_hasil];

      //Tambahkankehasilciphertext
      $pesan.=$bigram_hasil[$i];
}

      if(substr($pesan,strlen($pesan)-1,1)=="X")
      {
            //Buang karakter X yang ditambahkan sebelumnya pada
saatprosesenkripsi
            $pesan=substr($pesan,0,strlen($pesan)-1);
      }

      echo"
      <scripttype='text/javascript'>

      window.location='ekstraksi_langkah_3.php?pesan=$pesan'
;
      </script>
      ";
?>
```

LAMPIRAN B

DAFTAR RIWAYAT HIDUP

DATA DIRI

Nama	: Chyndy Astika Dani Hasibuan
Nim	: 0701162039
Tempat, Tanggal Lahir	: Kampung Lihas, 29 November 1998
Jenis Kelamin	: Perempuan
Alamat	: Jl. Sejarah Dusun IX
Kel/Desa	: Mekar Sari
Kecamatan	: Deli Tua
Kabupaten	: Deli Serdang
Agama	: Islam
Status Nikah	: Belum Menikah
Nama Orang Tua	
Ayah	: Sulaiman Hasibuan
Ibu	: Nuriani Purba

PENDIDIKAN FORMAL

2004-2010 : SD NEGERI 060928 MEDAN

2010-2013 : SMP NEGERI 34 MEDAN

2013-2016 : SMA NEGERI 4 TANJUNG BALAI

2016-2021 : UNIVERSITAS ISLAM NEGERI SUMATERA UTARA

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik 2019 / 2020

Nama	: CHYNDY AFTIKA DAWIH	Pembimbing I	: Dr. Mhd Furqan, S.Si, M.Comp. Sc
NIM	: 0701162039	Pembimbing II	: Yusup Ramadhan Nasution, M.Kom
Prog. Studi	: Ilmu Komputer	SK Pembimbing	:

Judul Skripsi :
IMPLEMENTASI KRIPTOGRAFI DALAM PENYISIPAN PESAN PADA CITRA DIGITAL MENGGUNAKAN METODE PLAYFAIR CIPHER DAN LEAST SIGNIFICANT BIT (LSB).

PERT	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	23/09/20	Bimbingan Bab 1		06/09/20	Bimbingan Bab 1 (Latar belakang, Rumusan masalah, Tujuan & manfaat)	
II	29/09/20	Revisi Bab 1		10/10/20	Bimbingan Bab 2 (Tinjauan Pustaka)	
III	5/10/20	Bimbingan Bab 2 dan Bab 3		21/09/20	Bimbingan Bab 3 (Cara kerja, dan perancangan).	
IV	15/10/20	ACC bab 1, dan 2		8/09/20	Revisi Bab 3, dan ACC bab 1 dan 2.	
V	18/10/20	ACC Proposal Skripsi		4/10/20	ACC Propocal Skripsi	

VI	17/04/2024	Bimbingan Bab IV		18/04/2024	Bimbingan Bab IV, Perbaiki.	
VII	18/04/2024	Perbaikan Bab IV		17/7/2024	Acc. Sidang	
VIII	19/04/2024	Acc Bab IV dan Bimbingan Bab V				
IX	20/04/2024	Acc Bab V				
X	22/04/2024	Acc Sidang		-		

Medan, ...20.....
An. Dekan
Ketua Jurusan/Program Studi

.....

NIP.

Catatan: *Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing*