

Penerapan Teknik Steganografi Untuk Keamanan Pengiriman Data Teks Ke Dalam Citra Berbasis Android

Madyana Patasik¹, Sri Wahyuni², Novita Sambo Layuk³

STMIK Dipanegara Makassar

Jalan Perintis Kemerdekaan KM. 9 Makassar, Telp. 041158719/ Fax. 0411588284

e-mail: madyanapatasik@gmail.com, sri.wahyuni@dipanegara.ac.id, fivhy@yahoo.co.id

Abstrak

Steganografi merupakan salah satu metode yang dapat digunakan untuk mengamankan informasi dengan menyembunyikan informasi ke media lain seperti citra digital, teks, suara atau video sehingga tidak menimbulkan kecurigaan orang lain. Media penampung yang banyak digunakan untuk menyembunyikan informasi yaitu citra digital. Algoritma *Lest Significant Bit* merupakan teknik steganografi dengan cara memodifikasi bit-bit yang termasuk bit *Lest Significant Bit* pada setiap byte warna pada sebuah pixel. Bit-bit *Lest Significant Bit* ini dimodifikasi dengan menggantikan setiap *Lest Significant Bit* yang ada dengan bit-bit informasi lain yang ingin disembunyikan. Penelitian ini berhasil membangun Aplikasi Steganografi yang dapat melakukan penyembunyian data ke dalam citra sebelum data tersebut dikirim melalui email berbasis android, dengan indikator aplikasi menghasilkan citra baru tanpa merusak citra asli, citra baru tersebut yang akan dikirim melalui media pengiriman data seperti email, maupun media transmisi data lainnya.

Kata Kunci : Steganografi , *Lest Significant Bit* , Citra Digital

Abstract

Steganography is one method that can be used to secure information by hiding information into other media such as digital images, text, sound or video so as not to arouse suspicion from others. Storage media that are widely used to hide information, namely digital images. The Lest Significant Bit algorithm is a steganography technique by modifying the bits that include the bit Lest Significant Bit on each color byte in a pixel. This bit-bit Lest Significant Bit is modified by replacing each existing Lest Significant Bit with other bits of information that you want to hide. This study succeeded in building a Steganography Application that can hide data into images before the data is sent via android-based e-mail, with application indicators producing new images without damaging the original image, the new image will be sent via data transmission media such as e-mail and transmission media other data.

Keywords: *Steganografi , Lest Significant Bit , Digital Image*

1. Pendahuluan

Steganografi adalah suatu teknik untuk menyembunyikan keberadaan pesan sehingga pesan yang dikirim tidak akan dicurigai mengandung pesan[1]. Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Hal ini mendorong keinginan untuk mencari metode alternatif dalam penyembunyian pesan yang tidak pernah terpikirkan sebelumnya.

Citra terbagi dua yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis[2].

Android memiliki aplikasi *Native Google* yang terintegrasi seperti *pushmail Gmail*, *Google Maps*, dan *Google Calendar*. Pengembang memiliki beberapa pilihan dalam membuat aplikasi yang berbasis Android. Namun kebanyakan pengembang menggunakan *Eclipse* dan *AndroidStudio* sebagai IDE untuk merancang aplikasi mereka. Hal ini dikarenakan mendapat dukungan langsung dari *Google* untuk menjadi pengembangan aplikasi Android[3].

Berdasarkan latar belakang di atas, maka penulis menganggap perlu dilakukan penelitian mengenai masalah tersebut dengan menyajikan penelitian dengan judul "Penerapan Teknik Steganografi untuk Keamanan Pengiriman Data Teks ke Dalam Citra Berbasis Android", maka diharapkan dapat menghasilkan sebuah aplikasi yang mampu memberi tanda digital ke file video.

2. Metode Penelitian

2.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penyusunan penelitian ini sebagai metode dalam pengumpulan data dan informasi antara lain:

1. Penelitian Lapangan (*Field Research*)
Yaitu penelitian yang dilakukan dengan cara melakukan pengamatan secara langsung kepada objek penelitian.
2. Penelitian Kepustakaan (*Library Research*)
Yaitu penelitian yang dilakukan dengan cara mengumpulkan data melalui beberapa sumber bacaan yang berkaitan dengan masalah yang diteliti dan bersifat ilmiah yang ada kaitannya dengan materi pembahasan.
3. Penelitian Eksperimental
Yaitu Penelitian yang dilakukan dengan cara menguji coba metode LSB untuk dapat digunakan dalam membangun aplikasi steganografi

2.1 Teknik Pengumpulan Data

Terdapat dua teknik pengumpulan data yang digunakan dalam penelitian ini, yaitu :

1. Teknik Observasi
Observasi atau pengamatan adalah suatu cara yang dilakukan untuk memperoleh data dengan mengamati fakta atau data yang digunakan oleh suatu organisasi tersebut.
2. Teknik Wawancara
Wawancara adalah suatu cara yang dilakukan untuk memperoleh jawaban atas pertanyaan yang berkaitan dengan masalah penelitian kepada bagian-bagian yang terkait di dalamnya.

2.2 Alat dan Bahan Penelitian

Selain menganalisis dan mengumpulkan data yang dibutuhkan dalam membangun aplikasi, selanjutnya dianggap perlu adanya alat dan bahan penunjang dalam penelitian yang dilakukan. Adapun alat dan bahan yang diperlukan antara lain :

2.2.1 Alat Penelitian

Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Software
 - a. Sistem Operasi *Windows 10 Home 64 Bit*, sistem operasi yang di gunakan dalam perancangan aplikasi ini.
 - b. *HTML5*, *Javascript*, *CSS3*, *PHP* sebagai bahasa pemrograman yang digunakan.
 - c. *Atom Editor*, editor yang di gunakan dalam penulisan *coding* aplikasi
 - d. *Laragon*, aplikasi pengganti *XAMPP* yang di gunakan untuk mendesain database pada aplikasi
2. Hardware
 - a. Laptop dengan *Processor Intel Core i7-7500U CPU 2.70GHz*, *harddisk 1 TB* dan *RAM 8 GB DDR4*, alat yang di gunakan dalam merancang aplikasi kami
 - b. Smartphone Android dengan sistem operasi *Android 6.0 (Marshmallow)*, *CPU Quad-core Max 1.4 GHz* dan penyimpanan internal *16GB*, alat yang kami gunakan dalam pengetesan dan pengaplikasian aplikasi yang kami rancang
3. Alat Desain
 - a. *Use Case Diagram* menggambarkan kelakuan (*behavior*) sistem secara keseluruhan yang akan dibuat.
 - b. *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.
 - c. *Activity Diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem.
 - d. *Sequence Diagram* menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

- e. *Entity Relationship Diagram (ERD)* menggambarkan hubungan yang ada dalam pengolahan data, seperti hubungan *many to many*, *one to many*, *one to one*.
- f. *State Transition Diagram (STD)* menggambarkan alur jalanya program.

Alat yang digunakan dalam penelitian berupa :

1. Perangkat Keras (Hardware)
Laptop intel(R) core(TM) i3 CPU M380 @ 2.53G.Hz
2. Perangkat Lunak (Software)
 - a. Windows 7 Ultimate 32-bit
 - b. Java Development Kit
 - c. NetBeans 7.0.1
3. Desain Konseptual

Dalam penelitian ini penulis menggunakan alat bantu dalam menganalisa dan mempelajari sistem yang dirancang. Alat bantu yang digunakan yaitu UML. Dalam perancangan sistem penulis menggunakan *Uses Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequential Diagram*.

2.2.2 Bahan Penelitian

Bahan yang digunakan dalam penelitian berupa perangkat lunak (*software*). Dalam melakukan perancangan sistem diperlukan beberapa file diantaranya :

1. File Citra (jpg)
2. File Teks (txt)

2.3 Teknik Pengujian Sistem

Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software. *Black box testing* merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*[4]. artinya teknik pengujian *black box* memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian ini bukan teknik alternatif untuk kotak putih, sebaliknya ini merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang diungkap oleh metode Kotak putih. Pengujian *black box* mengidentifikasi jenis kesalahan antara lain kesalahan suatu fungsi, kesalahan suatu antarmuka, kesalahan dalam pemodelan data dan kesalahan dalam akses ke sumber data eksternal[5]. Tidak seperti pengujian kotak putih, yang dilakukan pada awal proses pengujian, test *White Box* cenderung diterapkan selama tahap-tahap pengujian selanjutnya. Karena pengujian *Black Box* sengaja mengabaikan struktur kendali, perhatian difokuskan pada ranah informasi. Pengujian dirancang untuk menjawab pertanyaan – pertanyaan berikut : Bagaimana validasi fungsional di uji ?, Bagaimana perilaku dan kinerja sistem diuji ?, kelas-kelas masukan apakah yang akan membentuk test case yang baik ?, Apakah sistem sangat sensitif terhadap nilai masukan tertentu ?, Bagaimana batas-batas kelas data diisolasi ?, berapa kecepatan dan volume data yang dapat ditolerir oleh sistem ?, apa pengaruh kombinasi spesifik data pada operasi sistem ?.

2.4 Tahapan Penelitian

Tahapan yang harus dilalui dalam pembangunan sistem adalah sebagai berikut:

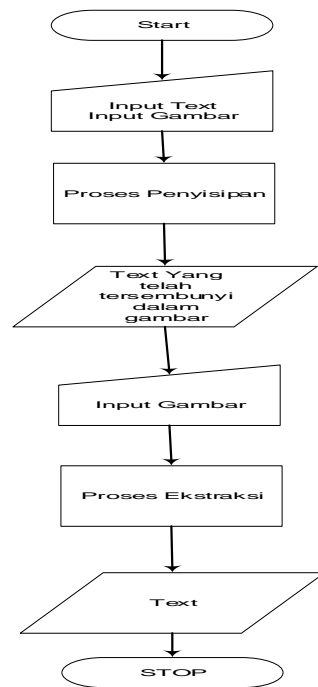
Pengumpulan data: Berupa pengumpulan data penunjang yang dapat membantu perancangan sistem.

1. Desain Logic: Desain Logic adalah pemilihan strategi arsitektur.
2. Pengkodean: Pengkodean adalah implementasi model ke dalam bahasa pemrograman.
3. Pengujian Perangkat Lunak: Pengujian perangkat lunak dilakukan setelah proses *coding* selesai untuk melakukan
4. Implementasi: Implementasi adalah abstraksi dari penerapan (implementasi) suatu sistem *software*.

3. Hasil dan Pembahasan

3.1 Rancangan Sistem Secara Umum

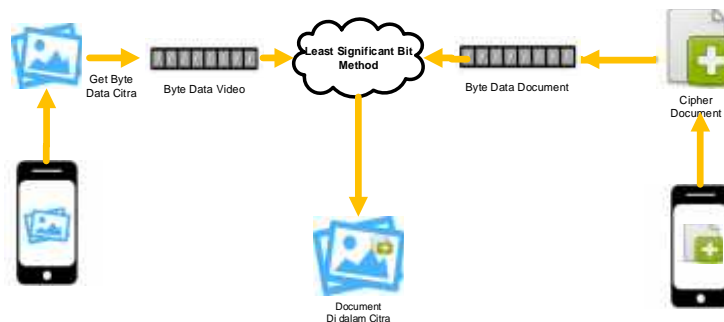
3.1.1 Flowchart Aplikasi



Gambar 1. Flowchart Aplikasi

3.1.2 Arsitektur Aplikasi

Langkah awal perancangan sistem secara umum dalam pembuatan aplikasi ini adalah membuat arsitektur aplikasi serta dokumentasi sistem dengan menggunakan UML (*Unified Modelling language*), dengan menggunakan beberapa buah diagram, yaitu : *use case diagram*, kemudian membuat *class diagram*, *sequence diagram* dan yang terakhir *activity diagram* yang menunjukkan setiap aktivitas pada setiap program. Berikut arsitektur aplikasi secara umum pada gambar 2 di bawah ini



Gambar 2. Arsitektur Aplikasi

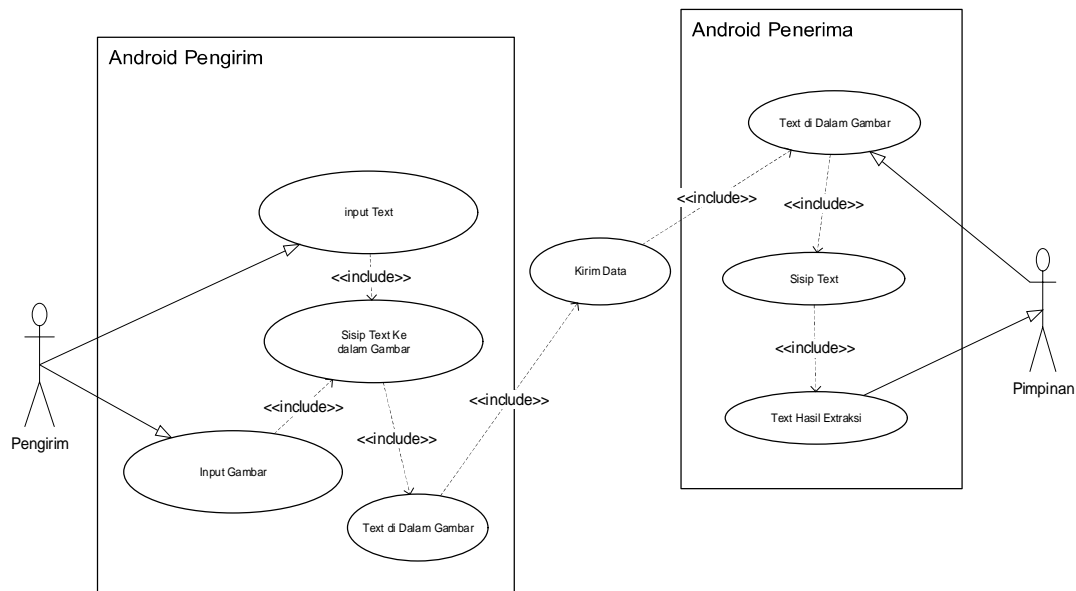
Pada gambar 2 di atas adalah arsitektur aplikasi yang akan dirancang, proses umumnya adalah meyembunyikan bit-bit teks pada bit-bit *Least* pada citra yang berfungsi sebagai media penampung teks.

3.1.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, *meng-create* sebuah daftar belanja, dan sebagainya.

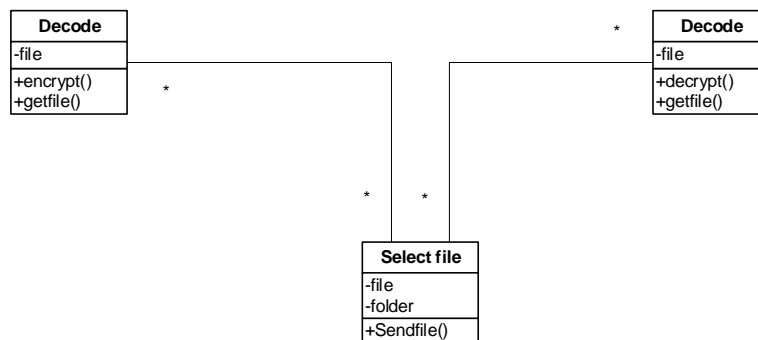
Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu apabila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem, adapun *use case* diagram pada penelitian ini dapat dilihat pada gambar 3 di bawah ini

1. Use Case Diagram



Gambar 3. Use Case Diagram

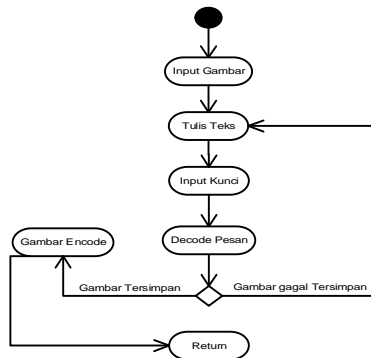
2. Class Diagram



Gambar 4. Class Diagram Aplikasi

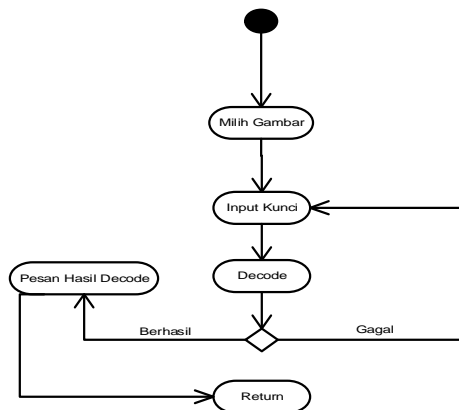
3. Activity Diagram

a. Activity Diagram Encode Pesan Email



Gambar 5. Activity Diagram Encode Teks ke dalam gambar

b. Activity Diagram Decode Teks dari Gambar

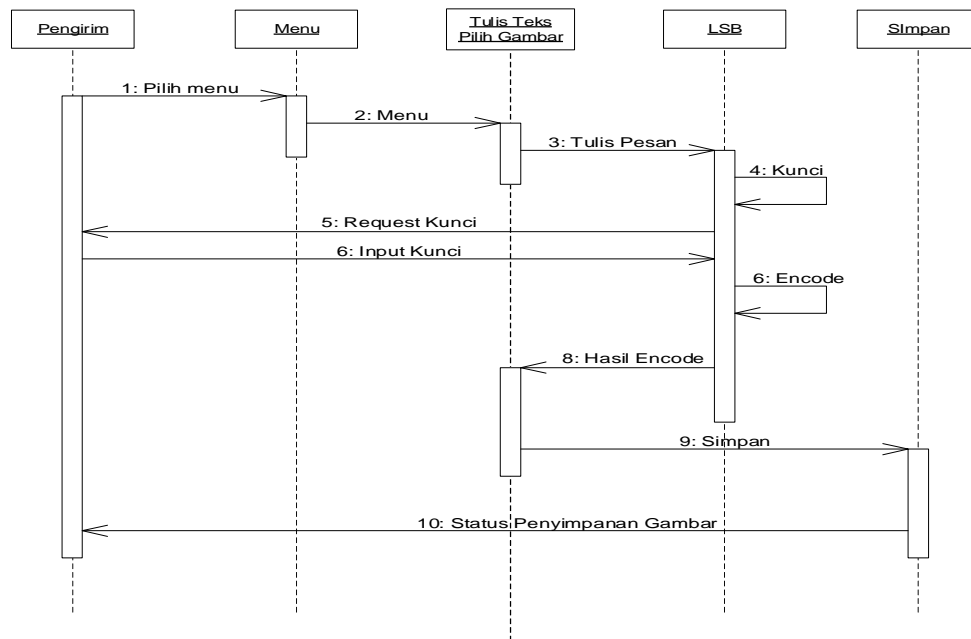


Gambar 6. Activity Diagram Decode Pesan Email

4. Sequence Diagram

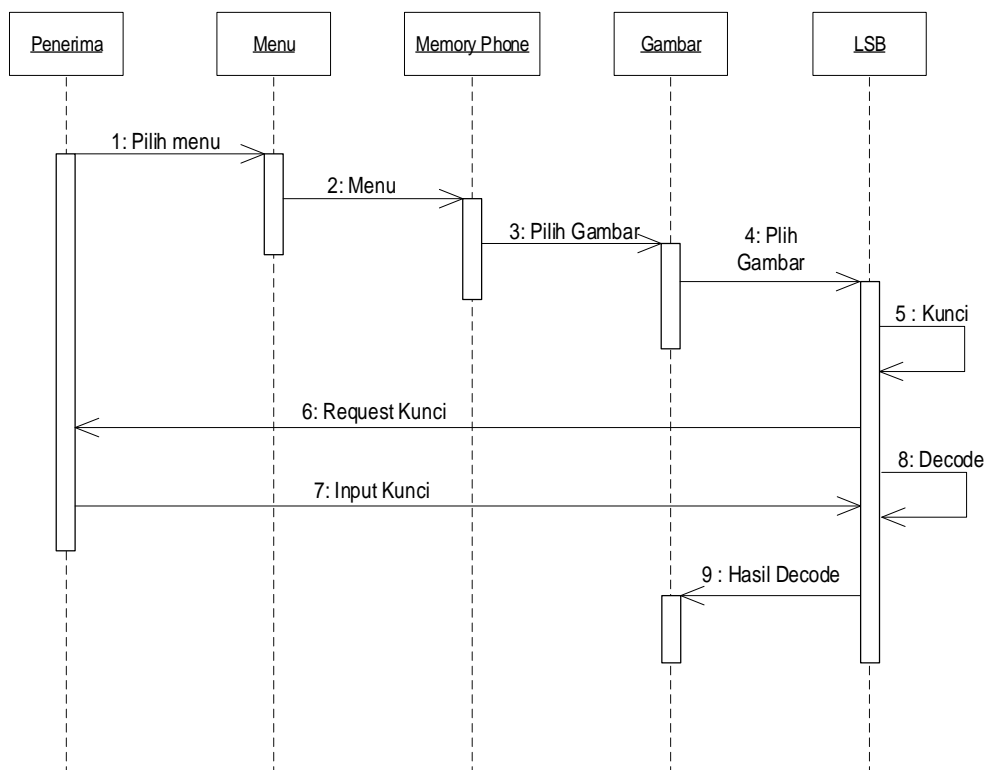
Sequence diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

a. Sequence Diagram Halaman User Encode



Gambar 7. Sequence Diagram Halaman Encode

b. Sequence Diagram Halaman User Decode



Gambar 8. Sequence Diagram Halaman Decode

5. Desain Antarmuka Output Input Secara Umum

a. *Output Menu Utama Aplikasi*



Gambar 9. *Output Menu Utama Aplikasi*

b. Antarmuka Encode



Gambar 10. Antarmuka Encode

c. Antarmuka Decode



Gambar 11. Antarmuka Decode

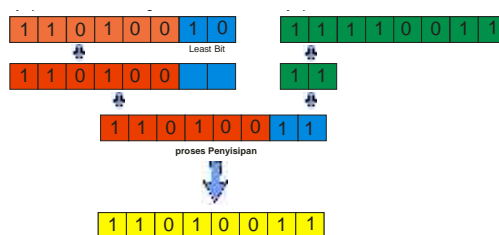
3.1.3 Implementasi Algoritma *Least Significant Bit*

1. Tahapan Proses Penyisipan

- Mengubah Data Citra dalam bentuk rangkaian biner.
- Mengubah Text dalam bentuk rangkaian biner.
- Sisipkan Data Text ke dalam Byte Citra Original dengan metode LSB.
- Buat File Citra Watermark dengan Data File yang telah diproses pada langkah 3.

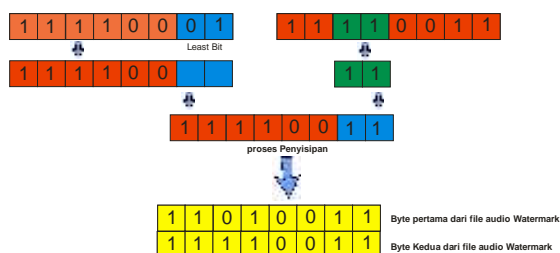
Gambar di bawah ini memperlihatkan proses penyisipan Citra dan Text dengan langkah-langkah di atas.

a. Skema byte pertama



Gambar 12. Skenario Penyisipan Byte Pertama

b. Skema byte kedua



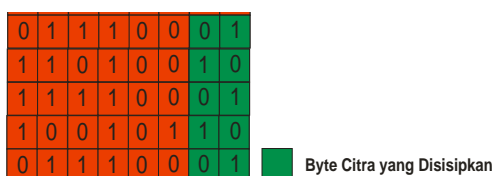
Gambar 13. Skenario Penyisipan Byte kedua

Skema diawali dengan mengosongkan dua bit pada byte pertama file Citra kemudian disiapkan dua bit pengganti yang diambil dari dua bit pertama dari byte file Citra dan Text, kemudian bit tersebut di tempatkan pada byte pertama yang telah dikosongkan sebelumnya, skema ini di ulang terus menerus sampai bit terakhir pada file Citra dan Text.

Adapun hasil perubahan nilai biner setelah penyisipan pada dua langkah di atas dapat dilihat pada gambar di bawah ini



Gambar 14. Hasil Perubahan byte



Gambar 15. Ilustrasi penempatan bit Citra dan Text.

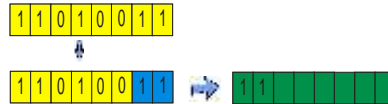
Gambar diatas menjelaskan bahwa dua bit terakhir pada byte ke adalah bit-bit dari file Citra dan Text yang disisipkan

2. Tahapan Proses Ekstraksi

- Mengubah Data Citra yang telah ber-*watermark* dalam bentuk byte data.
- Kembalikan Text yang telah disisipkan sebelumnya ke dalam data Citra watermark dengan metode LSB.
- Tampilkan Text dengan Data File Citra dan Text yang telah diproses pada langkah 2.

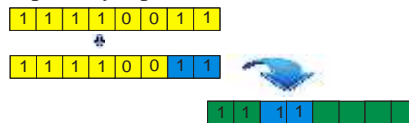
Gambar berikut ini memperlihatkan proses penyisipan Citra dan Text dengan langkah-langkah di atas

- Skema Ekstraksi Citra dan Text pada byte pertama file Citra



Gambar 16. Ekstraksi Citra dan Text pada byte pertama dari file Citra

- Skema Ekstraksi Citra dan Text pada byte pertama file Citra



Gambar 17. Ekstraksi Citra dan Text pada byte kedua dari file Citra watermark

Skema di atas menggambarkan tiap dua bit pada byte Citra diambil dan disatukan untuk membentuk deretan byte baru yang kemudian byte tersebut akan membentuk sebuah file Citra dan Text.

3. Perubahan Byte

Perhitungan perubahan byte dihitung berdasarkan selisih dari total nilai byte sebelum di sisipkan dan sesudah disisipkan.

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

(a)

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

(b)

Gambar 18. (a) Byte Original (b) byte Watermark

3.2 Pengujian Blackbox

1. Fungsi Pilih Citra Asli

Tabel 1. Pengujian Fungsi Pilih Citra Asli

Test Factor	Hasil	Keterangan
Memilhi Citra asli	✓	Dapat memilih Citra Asli dari Memory Handphone
Antarmuka		

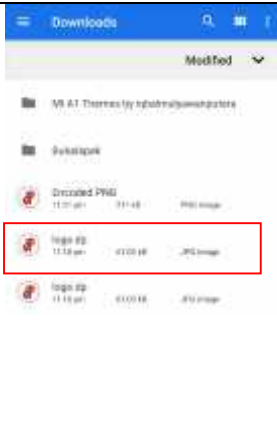
b. Fungsi Decode

Tabel 2. Pengujian Fungsi Decode Teks Ke dalam Gambar

Test Factor	Hasil	Keterangan
Menyembunyikan Teks Kedalam Gambar	✓	Menghasilkan Menyembunyikan Teks Kedalam Gambar
Antarmuka		
		

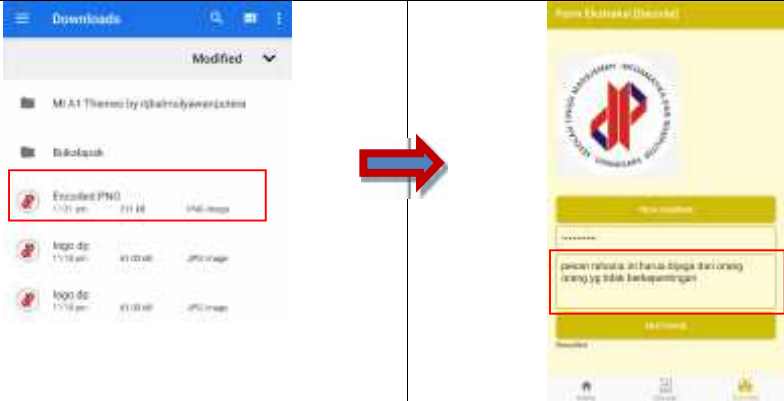
c. Fungsi Pilih Citra berisi teks

Tabel 3. Pengujian Fungsi Pilih Citra berisi teks

Test Factor	Hasil	Keterangan
Memilih Citra Berisi Teks	✓	Dapat memilih Citra berisi dari Memory Handphone
Antarmuka		
		

d. Fungsi Decode

Tabel 4. Pengujian Fungsi Decode Teks dari dalam gambar

Test Factor	Hasil	Keterangan
mengekstraksi Teks dari dalam gambar	✓	Sukses mengekstraksi Teks dari dalam gambar
Antarmuka		
		

Tabel 5. Rekapitulasi Hasil pengujian

No	Spesifikasi	Status	Hasil pengujian
1	Memilih Citra asli	✓	Dapat memilih Citra Asli dari Memory Handphone
2	Menyembunyikan Teks Kedalam Gambar	✓	Menghasilkan Menyembunyikan Teks Kedalam Gambar
3	Memilih Citra Berisi Teks	✓	Dapat memilih Citra berisi Teks dari Memory Handphone
4	mengekstraksi Teks dari dalam gambar	✓	Sukses mengekstraksi Teks dari dalam gambar

Dari tabel di atas dapat disimpulkan bahwa hasil keseluruhan pengujian input output dari aplikasi serta validasi aplikasi yang dibuat sudah sesuai dengan spesifikasi yang diinginkan, ini bisa dilihat dari keempat fungsi input output proses serta validasi sistem fungsional yang diinginkan dapat bekerja sesuai dengan spesifikasi yang diharapkan.

4. Kesimpulan

Berdasarkan hasil yang dicapai dari proses penelitian kami maka kami menarik kesimpulan bahwa :

1. Membangun Aplikasi Steganografi yang dapat melakukan penyembunyian data ke dalam citra sebelum data tersebut dikirim melalui email berbasis android, dengan indikator aplikasi menghasilkan citra baru tanpa merusak citra asli, citra baru tersebut yang akan dikirim melalui media pengiriman data. Data yang di terima akan dikembalikan ke naskah asli dengan melalui aplikasi yang telah dibangun.
2. Menerapkan metode Least Significant Bit Pada perangkat android untuk aplikasi steganografi, dengan mengambil tiap byte data pada teks, untuk disisipkan pada data terakhir tiap byte pada data citra.

Daftar Pustaka

- [1] Sentot, Kromodimoeljo, 2010, *Teori & Aplikasi Kriptografi*.SPK IT Consulting.
- [2] Hidayatullah, Priyanto, 2015, *Pengolahan Citra Digital; Teori dan Aplikasi Nyata*, Informatika, Bandung.
- [3] Nazaruddin, Safaat H, 2012, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Informatika, Bandung.
- [4] Mustaqbal M. Sidi, 2015, “*Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis*”, Universitas Widyatama, Bandung.
- [5] Hendraputra, Ade; Pratondo, Agus; Wijaya, Dedy Rahman, 2009, “*Jaminan Mutu Sistem Informasi*”, Politeknik Telkom, Bandung.