STEGANOGRAFI CITRA DIGITAL MENGGUNAKAN METODE LEAST SIGNIFICANT BIT (LSB)

Ervan Rahadian Hakim, Ida Astuti dan Winda Widya Ariestya
Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424
ervanrahadian@gmail.com, {astuti, winda_widya}@staff.gunadarma.ac.id

ABSTRAK

Penerapan steganografi dapat digunakan untuk menyembunyikan informasi dan mencegah pihak ketiga mengetahui keberadaan informasi. Tujuan penelitian ini adalah menghasilkan aplikasi steganografi menggunakan algoritma Least Significant Bit dan Cipher pada citra digitaldengan memberikan tambahan parameter kata kunci untuk memberikan keamanan. Metode SDLC (Software Development Life Cycle) dengan model waterfall dimanfaatkan dalam pengembangan sistem yang terdiri dari tahapan perencanaan, analisis, desain, implementasi dan pengujian. Pengujian dilakukan menggunakan metode black box testing dan kualitas steganografi. Hasil pengujiankualitas steganografi pada uji ketelitian (fidelity), format gambar (.png) lebih efisien dari segi perubahan ukuran gambar stegoyaitu rata-rata 20% dibanding dengan format gambar(.tiff) yaitu 42% dan format (.png) yaitu 83%. Pada uji pemulihan (recovery) diperoleh pesan teks yang disisipkan ke dalam citra digital dapat didekripsi dengan sempurna oleh program.

Kata Kunci : *Algoritma Cipher, Algoritma Least Significant Bit, Citra Digital, Steganografi.*

PENDAHULUAN

Pertukaran informasi melalui internet yang begitu maju, tidak dapat dipungkiri timbulnya kejahatan internet (*cybercrime*) seperti penyadapan, perubahan data dan lainnya. BerdasarkanDirektorat Tindak Pidana Siber Bareskrim Polri, terdapat 3.429 kasus tindak pidana siber dari Januari hingga Agustus 2019. Kejahatan *e-commerce* menjadi peringkat pertama, kejahatan tersebut berupa pencurian alamat *e-mail*, *username*, *password* dan PIN yang disalahgunakan oleh pihaklain. Kondisi demikian dibutuhkan suatu teknik yang dapat melindungi informasi tersebut salah satunya adalah teknik steganografi (Dedi Darwis, 2016).

Steganografi adalah seni menyembunyikan informasi untuk mencegah pihak ketiga mengetahui keberadaan pesan, sehingga steganografi dapat diartikan sebagai tulisan yang terselubung. Pesan rahasia dapat disisipkan dalam media berupa teks, citra, *audio* dan *video*. Penyembunyian pesan rahasia dalam citra yang mengandung informasi tersembunyi disebut *stego image*, sedangkan citra yang tidak mengandung informasi tersembunyi disebut *cover image* (Munir, 2009).

Salah satu algoritma steganografi yang bisa digunakan untuk menyembunyikan pesan adalah LSB (*Least Significant Bit*). Algoritma LSB adalah teknik subtitusi yang membuat proses steganografi menjadi efektif dan tidak terlalu mengubah ukuran data asli(Ariyus, 2009). Beberapa penelitian terdahulu mengenai pemanfaatan algoritma LSB pada steganografi diantaranya olehDarwis (2016) LSB diimplementasikan pada pengamanan data pengiriman surat elektronik tetapi belum menggunakann parameter kunci (*password*). Farid dkk (2016) memanfaatkan LSB untuk menyembunyikan pesan rahasia dan melakukan analisa kualitas citra stego diperoleh hasil citra stego tidak dapat diekstrak dengan baik sehingga *plaintext* tidak sesuai.Anwar (2016) algoritma LSB diimplementasikan untuk pengamanan data dan informasi dengan menampilkan statistic waktu pemrosesan tetapi format citra digital yang digunakan terbatas pada (.bmp). Nurdiansyah dan Ayu (2017) steganografi citra digital menggunakan LSB diimplementasikan pada pemberkasan arsip tetapi gambar hasil stego tidak dapat dikembalikan pesan rahasia di dalamnya.

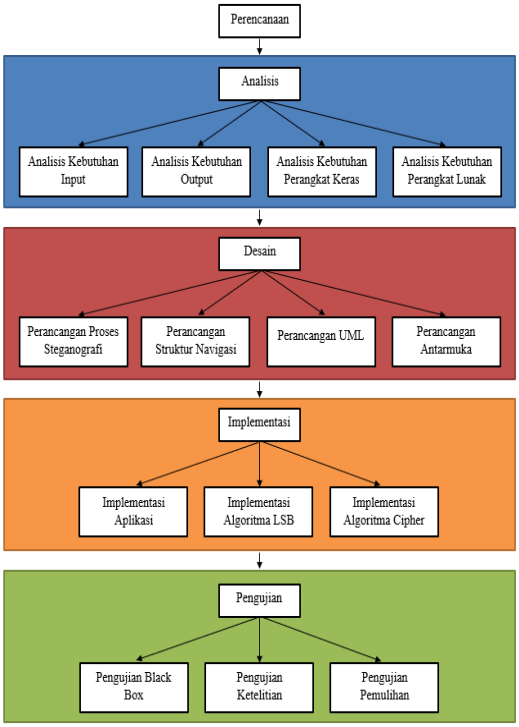
Berdasarkan latar belakang tersebut, penelitian ini memanfaatkan algoritma

cipher untuk proses enkripsi dan dekripsi dan algoritma LSB dalam steganografi citra digital dengan melakukan pengembangan pada pembuatan parameter kunci (*password*) untuk melindungi pesan yang disisipkan ke dalam citra digital dan dilakukan pengujian terhadap*fidelity* dan *recovery*.

METODE PENELITIAN

Penelitian ini menggunakan algoritma LSB karena ukuran informasi yang disisipkan tidak terjadi perubahan pada media asli (Hafiz, 2019). Pada LSB pergantian *bit* tidak begitu berpengaruh, hanya berubah satu tingkat dari *byte* sebelumnya. Penyembunyian pesan dilakukan dengan cara mengganti *bit* terendah pada media dalam hal ini citra digital. Pesan yang akan disembunyikan diubah ke dalam bentuk biner, kemudian disisipkan pada citra digital dengan menggunakan algoritma LSB. Penyisipan pesan ke dalam *cover image* harus diperhatikan tingkat kerahasiaan pesan yang akan disisipkan. Pesan rahasia sebaiknya terlebih dahulu dilakukan enkripsi, agar tidak mudah terbaca.

SDLC (*Software Development Life Cycle*) dengan model *waterfall*.digunakan sebagai tahapan pengembangan sistem seperti tertera pada Gambar 1 yang terdiri dari tahap perencanaan, analisis, desain, implementasi dan pengujian(Alshamrani dan Bahattab, 2015).

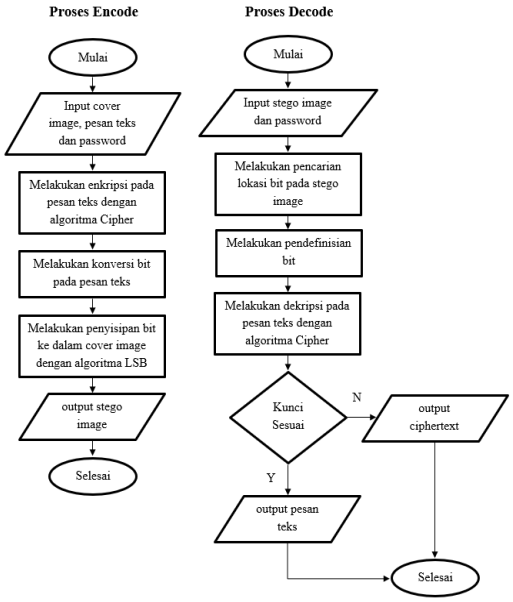


Gambar 1. *Metode Penelitian*

HASIL DAN PEMBAHASAN

Perancangan algoritma LSB dan *Cipher* ini terdiri dari proses *Encode* untuk penyisipan pesan teks ke dalam *cover image* dan proses *Decode* untuk mendapatkan pesan teks yang tersembunyi dalam *stego image* seperti pada Gambar 2.

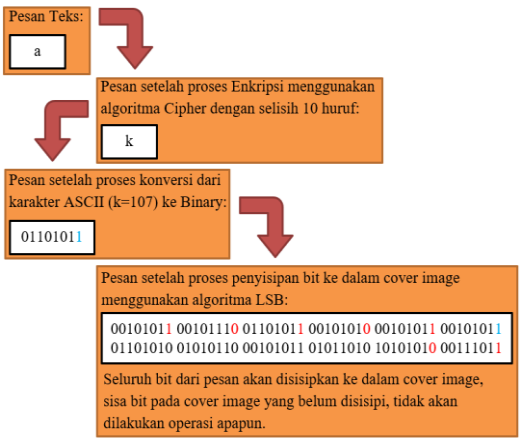
Gambar 3 adalah aktivitas proses *Encode*yang dimulai dengan memasukan *cover image*, pesan teks dan *password*, kemudian dilakukan enkripsi pada pesan teks dengan algoritma *Cipher*, lalu dilakukan konversi *bit* pada pesan teks. Pesan teks yang sudah dikonversi menjadi *bit* kemudian disisipkan ke dalam *cover image* dengan algoritma LSB dan akan dihasilkan *stego image*.



Gambar 2. *Proses Steganografi*

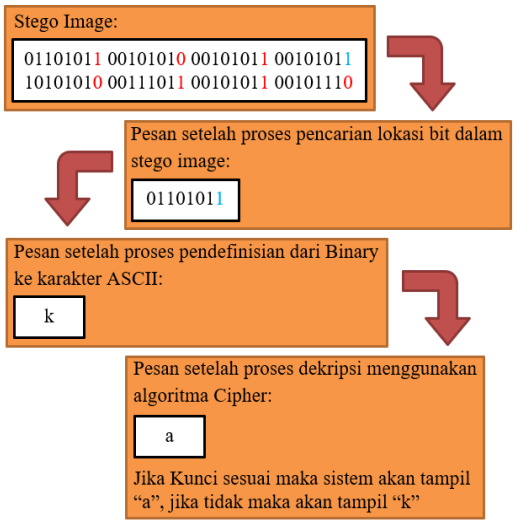
steganografi yaitu proses *encode*(memasukkan gambaratau*cover image*, memasukkan pesan teks, memasukkan *password* dan memberi nama gamabar atau *stego image*), serta proses *decode*(memasukkan gambar atau *stego image* dan memasukkan *password)*. Pengguna juga melihat form teori, melihat form cara penggunaan, melihat form tentang dan keluar aplikasi.

Aplikasi steganografi ini menggunakan algoritma *Least Significant Bit* (LSB) untuk penyisipan pesan teks digital ke dalam citra digital. Sebelum pesan dapat disisipkan ke dalam citra digital



Gambar 3. *Proses Encode*

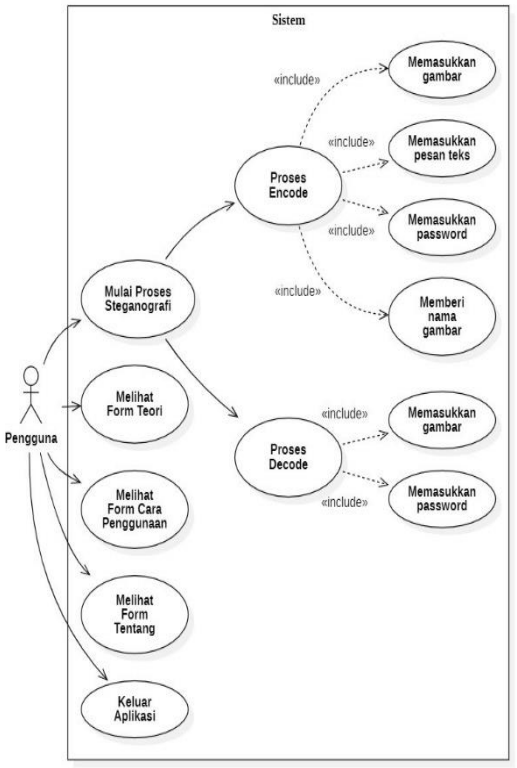
Gambar 4. Adalah aktivitas proses *Decode* yang dimulai dengan memasukan *stego image* dan *password*, kemudian dilakukan pencarian lokasi bit pada stego image, lalu dilakukan pendefinisian *bit* pada *stego image* untuk menemukan pesan teks. Pada teks yang terenkripsi dilakukan dekripsi dengan algoritma *Cipher*, jika *password* sesuai dengan identitas *stego image* maka pesan teks akan ditampilkan, jika tidak sesuai maka pesan terenkripsi akan ditampilkan.



Gambar 4. *Proses Decode*

Gambar 5 menggambarkan sistem dari sudut pandang pengguna yang melakukan interaksi dengan sistem. Interaksi ini terdiri dari mulai proses

, perlu dilakukan konversi *bit* (*bit conversion*) pada pesan. Konversi ini dibutuhkan agar satuan pada pesan teks digital dan citra digital sama sehingga dapat dilakukan penyisipan.



Gambar 5. *Use Case Diagram*

Dalam *bit conversion* 1 *byte* = 8-*bit*, yang berarti *byte* memiliki kelipatan dari *bit*. *Bit conversion* ini dapat diimplementasikan ke dalam sistem.Gambar 6 adalah potongan *source code* untuk *bit conversion*.

Berikutnya dilakukan *embedding*, yaitu bagaimana menyisipkan *bit* tersebut ke dalam sebuah citra digital.

Proses selanjutnya adalah *extracting* yaitu proses pembuktian aplikasi berjalan dengan baik, maka sistem harus bisa mendapatkan kembali pesan teks digital yang sudah disisipkan.

```
{
int fp=0,lp=32,j=0,i,c=0;
BigInteger encod[]=new BigInteger[1000000];
BigInteger encry[]=new BigInteger[1000000];
String binary="";
String coded=new String();
String sixteen[]=new String[1000000];
String str="";
for(i=0;i<msg1.length();i++)
}
```

Gambar 6.*Source Code Bit Conversion*

Selain algoritma *Least Significant Bit* (LSB), aplikasi steganografi ini juga menggunakan algoritma *Cipher*. Algoritma *Cipher* disini digunakan untuk proses enkripsi dan dekripsi pada pesan teks digital yang akan disisipkan ke dalam citra digital. Enkripsi merupakan proses mengamankan informasi dengan membuat informasi di dalam pesan teks digital tidak dapat dibaca tanpa kunci (*key*) khusus. Gambar 7 adalah potongan *source code* untuk enkripsi.

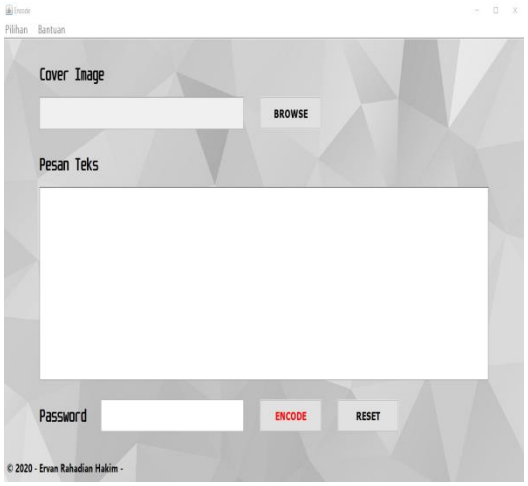
```
public String Enkrip (String plain, String key){
    plain(plain);
    key(key);
    String enkrip;
    char[] x = new char[Plain.length];
    for (int i = 0; i < Plain.length; i++) {
        x[i] = printChar((Plain[i]) % 256);
    }
    enkrip = String.valueOf(x);
    System.out.println(enkrip);
    return enkrip;
}
```

Gambar 7.*Source Code* Enkripsi

Selanjutnya adalah proses dekripsi, dekripsi merupakan kebalikan dari enkripsi yaitu proses mendapatkan kembali informasi dengan menggunakan kunci (*key*) khusus.

Implementasi metode LBS dan Cipher pada penelitian ini ditampilkan pada Gambar 8. Pengguna akan diberikan pilihan beberapa menu dai aplikasi untuk menjalankan aplikasi. Proses *encode* diimplementasikan pada halaman *encode* Gambar 9.

Gambar 8. *Tampilan Halaman Aplikasi*



Gambar 9. *Tampilan Halaman Encode*

Pengujian *black box*yaitu evaluasi sistem dari tampilan *interface* dan fungsional tanpa mengetahui proses secara keseluruhan, pengujian ini dilakukan dengan melalui 13 skema pengujian seperti pada Tabel 1.

Tabel 1. *Pengujian Black Box*

Fungsional	Target	Hasil	Keterangan
ButtonMulai.	Dapat menampilkan Form *Encode*.	Dapat menampilkan Form *Encode*.	Berhasil
Menu *Encode*.	Dapat menampilkan Form *Encode*.	Dapat menampilkan Form *Encode*.	Berhasil
Button *Browse Cover Image*.	Dapat mencari dan memilih *cover image*.	Dapat mencari dan memilih *cover image*.	Berhasil
Button *Encode*.	Dapat melakukan proses *embedding*.	Dapat melakukan proses *embedding*.	Berhasil
Menu *Decode*.	Dapat menampilkan Form *Decode*.	Dapat menampilkan Form *Decode*.	Berhasil
Button *Browse Stego Image*.	Dapat mencari dan memilih *stego image*.	Dapat mencari dan memilih *stego image*.	Berhasil
Fungsional	**Target**	**Hasil**	**Keterangan**
Button *Decode*.	Dapat melakukan proses *extracting*	Dapat melakukan proses *extracting*	Berhasil
Button *Image Preview*.	Dapat menampilkan citra digital.	Dapat menampilkan citra digital.	Berhasil
Button *Reset*.	Dapat membersihkan data yang terdapat pada Form.	Dapat membersihkan data yang terdapat pada Form.	Berhasil
ButtonTeori.	Dapat menampilkan Form Teori.	Dapat menampilkan Form Teori.	Berhasil
ButtonCara Penggunaan.	Dapat menampilkan Form Cara Penggunaan.	Dapat menampilkan Form Cara Penggunaan.	Berhasil
ButtonTentang.	Dapat menampilkan Form Tentang.	Dapat menampilkan Form Tentang.	Berhasil
ButtonKeluar.	Dapat menutup aplikasi.	Dapat menutup aplikasi.	Berhasil

Dilakukan pengujian ketelitian (*fidelity*) pada steganografi yang dihasilkan berdasarkan penelitian oleh Dedi Darwis (2016), pengujian ini dilakukan untuk mengetahui besarnya perubahan citra digital setelah disisipi pesan teks. Hasil pengujian ditunjukan pada Tabel 2, diperoleh hasil bahwa format gambar (.png) terjadi perubahan ukuran rata-rata sebanyak 20%, (.tiff) terjadi perubahan ukuran rata-rata 42% dan pada format (.jpg) terjadi perubahan ukuran rata-rata sebesar 83%, dapat dikatakan bahwa penggunaan jenis format gambar (.png) dan (.tiff) lebih efisien dari segi perubahan ukuran akhir dibandingkan dengan format (.jpg) dengan jumlah karakter pesan teks sebanyak 3605 karakter.

Pengujian berikutnya yaitu pemulihan (*recovery*) pesan teks pada steganografi yang dihasilkan oleh aplikasi ini, pengujian ini dilakukan untuk mengetahui apakah pesan teks yang disembunyikan pada *stego image* dapat didekripsi kembali.Diperoleh hasil uji pemulihan bahwa pesan teks yang disisipkan ke dalam citra digital dapat didekripsi dengan baik oleh program pada 3605pesan teks dan pesan teks hasil deskripsi seperti yang tertera pada Tabel 3.

Tabel 2.*Pengujian Ketelitian (Fidelity) Steganografi*

Cover Image	Ukuran Cover Image	Stego Image	Ukuran Stego Image
Gambar1.png	271 KB	Stego1.png	447 KB
Gambar2.png	178 KB	Stego2.png	227 KB
Gambar3.tiff	85,3 KB	Stego3.png	149 KB
Gambar4.tiff	89,7 KB	Stego4.png	153 KB
Gambar5.jpg	291 KB	Stego5.png	1,33 MB
Gambar6.jpg	234 KB	Stego6.png	1,84 MB

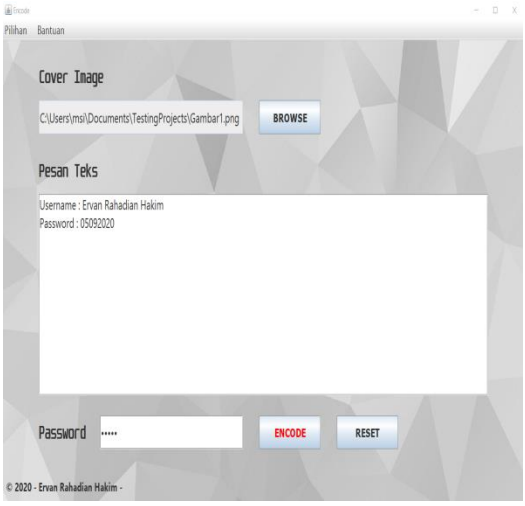
Tabel 3.*Pengujian Pemulihan (Recovery)*

Cover Image	Stego Image	Keterangan
Gambar1.png	Stego1.png	Berhasil
Gambar2.png	Stego2.png	Berhasil
Gambar3.tiff	Stego3.png	Berhasil
Gambar4.tiff	Stego4.png	Berhasil

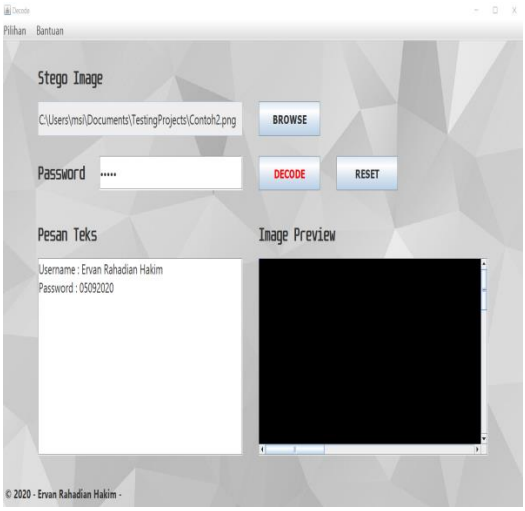
Gambar5.jpg	Stego5.png	Berhasil
Gambar6.jpg	Stego6.png	Berhasil

Gambar 10 merupakan contoh pengujian pemulihan (*recovery*), langkah pertama adalah melakukan *input* data pada *form encode* untuk proses penyisipan pesan teks ke dalam citra digital. Data yang dimasukkan adalah *cover image*, pesan teks yang berisi 50 karakter dengan *password* yang berisi "12345".

Selanjutnya dilakukan penyimpanan file. Berikutnya lakukan*input* data pada form *decode*seperti pada Gambar 11. Data yang dimasukkan adalah *stego image* dan *password* untuk verifikasi, klik Button *Decode*, maka pesan teks hasil proses *Decode* akan ditampilkan pada Text Area.

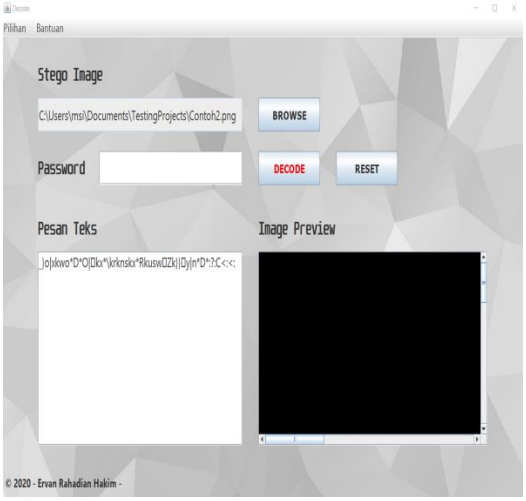


Gambar 10.*Input Encode*



Gambar 11.*Input Decode* Benar

Gambar 12 adalah tampilan yang terjadi jika *password* yang dimasukkan salah dan sistem akan menampilkan pesan teks dengan enkripsi.



Gambar 12 *Input Decode* Salah

PENUTUP

Pengembangan aplikasi steganografi dengan menambahkan parameter kunci (*password*) untuk menambah keamanan berhasil dilakukan dengan mengimplementasikan Algoritma LSB dan *Cipher* untuk penyisipan teks ke dalam citra digital.Pengujian *black box*pada aplikasi berjalan sesuai dengan fungsi. Pengujian kualitas yang terdiri dari uji ketelitian (*fidelity*) dihasilkan format gambar (.png) lebih efisien dari segi perubahanukuran gambar stegoyaitu rata-rata 20% dibanding dengan format gambar(.tiff) yaitu 42% dan format (.png) yaitu 83%.Hasil pengujian pemulihan (*recovery*)dihasilkan pesan teks yang disisipkan ke dalam citra digital dapat didekripsi dengan baik oleh program.

Padapengembangan selanjutnya dapat ditambahkan pesan rahasia berupa *image*, *audio* dan *video*, kemudian media sampul (*cover*) berupa *audio* dan *video*.

DAFTAR PUSTAKA

[1] A. Alshamrani dan A. Bahattab, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model And Incremental/Iterative Model",*inInternational Journal of Computer Science Issues (IJCSI)*,vol. 12, no.1,pp. 106, 2015.

[2] A. Hafiz, "Steganografi Berbasis Citra Digital Untuk Menyembunyikan Data Menggunakan Metode Least Significant Bit (LSB)", in *Jurnal Cendikia*, vol. 17, no. 1, pp. 194-198, April 2019.

[3] D. Ariyus,*Keamanan Multimedia*, Yogyakarta: Andi Offset, 2006.

[4] D. Darwis,"Implementasi Teknik Steganografi Least Significant Bit (LSB) dan Kompresi Untuk Pengamanan Data Pengiriman Surat Elektronik, in*Teknoinfo*, vol. 10, no. 2, pp. 1-7, 2016.

[5] N. Farid, B. Nurhadiyono, dan Y. Rahayu, "Implementasi Metode Steganografi Least Significant Bit Dengan Algoritma Hill Cipher Pada Citra Bitmap", in*Techno. Com*, vol. 15, no.2,pp. 109-116, 2016. DOI: https://doi.org/10.33633/tc.v15i2.1145

[6] R. Munir, *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*, Bandung: Informatika, 2004.

[7] S. Anwar, "Implementasi Pengamanan Data Dan Informasi Dengan Metode Steganografi LSB Dan Algoritma Kriptografi AES", in *Semnasteknomedia Online*, vol. 5, no.1, pp. 3-8, 2017.

[8] Y. Nurdiansyah dan A.L.F. Riftana, "Implementasi Steganografi Citra Digital Pemberkasan Arsip Menggunakan Metode Least Significant Bit",in Seminar Nasional Informatika dan Aplikasinya (SNIA) 2017Cimahi, ISBN: 978-602-50525-0-7, pp. C2-C7, 27 September 2017.