Volume 1 Nomor 1, April 2018
e-ISSN : 2621-234X

Implementasi Steganografi Menggunakan Algoritma *Diversity* Pada Citra Digital

Roji Ramadhani[1*], Dodi Siregar[1], Yunita Sari Siregar[1]

Address : Program Studi Teknik Informatika, Universitas Harapan Medan, Medan, Indonesia

Email : Rojirama12345@gmail.com

Correspondence *

Abstract

Abstract- Providing security and confidentiality to information is very important when exchanging information though a communication network. It is intended that the information sent by the sender can be fully accepted by the recipient without interference from parties who are not interested in the information. Steganography techniques can be used to secure secret massages. Steganography is a method for hiding information on a media. Can be in the form of image, sound or video media. The most important aspect of steganography is the level security in hiding information, which refers to how much the third party is unable to detect hidden information. The goal is to avoid suspicion. Steganography commonly used is the hiding of text information on image media. The method used is Diversity. From the results of the trial, it is known that with Diversity insertion and extraction of massages can be done well.

Keywords- Steganography, Diversity, Text.

Abstrak

Memberikan keamanan dan kerahasiaan pada informasi sangat penting ketika bertukar informasi melalui jaringan komunikasi. Hal ini dimaksudkan bahwa informasi yang dikirim oleh pengirim dapat diterima oleh penerima tanpa campur tangan dari pihak yang tidak berkepentingan terhadap informasi tersebut. Teknik steganografi dapat digunakan untuk mengamankan pesan rahasia. Steganografi adalah metode untuk menyembunyikan informasi pada sebuah media, bisa berupa media gambar, suara ataupun video. Aspek terpenting pada steganografi adalah tingkat keamanan penyembunyian informasinya, yang megacu pada seberapa besar ketidakmampuan pihak ketiga dalam mendeteksi keberadaan informasi yang tersembunyi. Tujuannya untuk menghindari kecurigaan. Steganografi yang umumnya digunakan adalah penyembunyian informasi teks pada media gambar. Metode yang dipakai adalah *Diversity*. Dari hasil uji coba, diketahui bahwa dengan *Diversity* penyisipan dan ekstraksi pesan dapat dilakukan dengan baik.

Kata kunci- Steganografi, *Diversity,* Teks.

1. Pendahuluan

Perkembangan teknologi digital serta internet saat ini telah memberi kemudahan untuk melakukan akses serta mendistribusikan berbagai informasi dalam format digital. Beberapa faktor yang membuat data digital (suara, citra, video dan teks) banyak digunakan antara lain dikarenakan kemudahan dalam proses duplikasi dan hasil dari duplikasi akan tetap sama dengan aslinya, juga dikarenakan biaya yang murah dalam proses duplikasi dan penyimpanan, serta

mudah disimpan dan kemudian untuk diolah atau diproses lebih lanjut.

Steganografi merupakan salah satu metode yang dapat digunakan untuk mengamankan pesan rahasia pada media lain seperti (citra, audio, video maupun teks). Ada beberapa metode yang digunakan dalam steganografi antara lain metode *least significant bit end of file*. Kualitas citra pada metode *least significant bit* dapat diukur dengan melakukan perhitungan nilai *Mean Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR), sedangkan kualitas citra berdasarkan metode *and of file* tidak dapat dihitung dengan perhitungan MSE dan PSNR karena kualitas citra awal dan citra setelah disisipkan pesan tidak mengalami perubahan[1].

Steganografi yang merupakan kombinasi dari tiga warna dasar *Red, Green, Blue* (RGB). Sedangkan satu *pixel* citra warna 24 bit diwakili oleh tiga *byte,* dimana masing-masing byte merepresentasikan warna *Red, Green* dan *Blue*. Penyisipan pesan ke dalam *cover image* dinamakan *encoding*, sedangkan ekstraksi pesan dari stego image dinamakan *decoding*. Kedua proses memerlukan kunci rahasia (*stego key*), agar hanya pihak yang mempunyai kunci rahasia saja yang dapat melakukan penyisipan dan ekstraksi pesan. Proses *encoding* dan *decoding*[2].

Perubahan yang terjadi pada citra stegano hanyalah besar data sebelum dan sesudah proses steganografi, namun tidak merubah bentuk dari citra dalam skala yang besar[3].

2. Metode Penelitian

2.1 Citra

Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. [4].

Steganografi

Steganografi adalah ilmu seni menulis atau menyembunyikan pesan ke dalam sebuah media sehingga keberadaan pesan tidak dapat diketahui atau tidak disadari oleh indera manusia. Teknik steganografi berfungsi menyembunyikan pesan rahasia di dalam media digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang lain[6].

2.3 Teknik Penyembunyian Data

Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia. Salah satu metode penyembunyian data yang sederhana adalah *LSB Modification*. Perhatikan contoh penyembunyian data yang sederhana untuk susunan pada sebuah *byte,* sebagai berikut:

1 1 0 1 0 0 1 0
↓ ↓ MSB = *Most Significant Bit.*
MSB LSB
LSB = *Least Significant Bit.*

Bit yang cocok untuk diganti adalah bit *LSB*, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut menyatakan warna keabuan tertentu, maka perubahan satu bit LSB tidak mengubah warna keabuan tersebut secara berarti. Secara kasat mata manusia tidak dapat membedakan perubahan yang kecil di dalam perubahan warna yang terseleksi pada citra. Misalkan segmen data citra sebelum perubahan, seperti contoh berikut:

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

Segmen data citra setelah " 1 0 0 1 " yang disembunyikan :

↓ ↓ ↓ ↓
0 0 1 1 0 0 1 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 1 1 1

Untuk memperkuat teknik penyembunyian data, bit-bit data rahasia tidak digunakan mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka maka *byte* yang diganti bit *LSB*-nya dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49. Bilangan acak dibangkitkan dengan *pseudo-random-number-generator* (*PRNG*) kriptografi.*PRNG* kriptografi sebenarnya adalah algoritma kriptografi yang digunakan untuk enkripsi.*PRNG* dibangun dengan algoritma *DES* (*DataEncryption Standard*), algoritma *hash MD5*, dan mode kriptografi *CFB* (*Chiper-Feedback Mode*). Tujuan enkripsi adalah menghasilkan

sekumpulan bilangan acak yang sama untuk setiap kunci enkripsi yang sama. Bilangan acak dihasilkan dengan memilih bit-bit dari sebuah blok data hasil enkripsi. Teknik penyembunyian data untuk citra 8-bit berbeda dengan citra 24-bit. Citra *bitmap* terdiri atas bagian *header*, palet *RGB*, dan data *bitmap*. Pada citra 8-bit, setiap elemen data *bitmap* menyatakan indeks dari peta warnanya di palet *RGB*[3]. Perhatikan elemen data *bitmap* menyatakan indeks dari peta warnanya di palet *RGB*, seperti contoh berikut:
Format citra 8-bit (256 warna):

< *header*>
< palet *RGB* >

	R	*G*	*B*
1	20	45	24
2	14	13	16
3	12	17	15
. . .			
256	46	78	25

< data *bitmap* >
2 2 1 1 1 3 5 . . .

Pada citra 24-bit, tidak terdapat palet *RGB*, karena nilai *RGB* langsung diuraikan dalam data *bitmap*.Setiap elemen data *bitmap* panjangnya 3 *byte*, masing-masing *byte* menyatakan komponen *R*, *G*, dan *B*. Perhatikan seperti contoh berikut:
Format citra 24-bit (16 juta warna):
< *header* >
< data *bitmap* >
2 2 1 1 1 3 5 . . .
Pada contoh format citra 24-bit di atas, *pixel* pertama mempunyai *R* = 2, *G* = 2, *B* = 1.

3 Analisis Permasalahan

Masalah dari penelitian ini adalah pembuatan sistem steganografi pada pesan teks pada citra digital menggunakan metode *diversity*. Metode *diversity* merupakan metode yang digunakan untuk mensubstitusikan bit-bit dari pesan teks dengan hasil transformasi bit piksel dari citra penampung. Steganografi pada citra digital menggunakan metode *diversity* pada citra digital untuk mencari hasil transformasi. Penerapan metode tersebut pada steganografi menghasilkan proses steganografi yang memiliki daya tampung yang lebih besar dengan berupaya menjaga kualitas dari citra penampung.

3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem meliputi analisis kebutuhan fungsional dan analisis kebutuhan *non*-fungsional sistem. Kebutuhan fungsional sistem adalah kebutuhan yang berisi proses-proses atau layanan apa saja yang nantinya harus disediakan oleh sistem, sedangkan kebutuhan *non*-fungsional sistem adalah kebutuhan yang menitik beratkan pada properti prilaku yang dimiliki oleh sistem. Adapun kebutuhan fungsional dan *non*-fungsional sistem yang harus dimiliki oleh sistem steganografi pada citra digital menggunakan metode *diversity* adalah,sebagai berikut :

1. Analisi kebutuhan fungsional sistem :
   a. Sistem dapat melakukan ekstraksi piksel – piksel dari citra penampung.
   b. Sistem dapat membentuk matriks – matriks transformasi dari tiap piksel yang digunakan pada proses substitusi.
   c. Sistem dapat mencari matriks transformasi yang memiliki nilai antara RSME pada saat proses substitusi dengan bit pesan.
   d. Sistem dapat melakukan penyisipan dan ekstraksi kembali.
2. Analisis Kebutuhan *Non*-Fungsional Sistem :
   a. Sistem memiliki proses yang akurat.

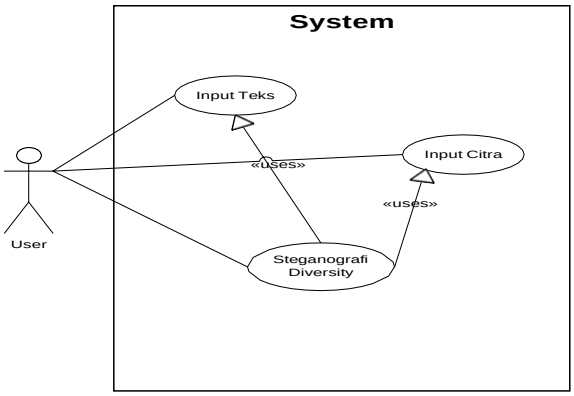
Tampilan antarmuka sistem menarik dan dapat dimengerti oleh pengguna sistem.

3.2 Pemodelan

Penelitian ini menggunakan UML sebagai bahasa pemodelan untuk mendesain dan merancang sistem steganografi penyembunyian pesan teks pada citra digital menggunakan metode algoritma *diversity*. Model UML yang akan digunakan adalah *use case* dan *activity diagram*.

3.3 pemodelan Menggunakan *Use Case Diagram*

Untuk mengetahui aktor dan *use case* yang akan digunakan, maka dilakukan identifikasi aktor dan identifikasi *use case*. Setelah mendapatkan aktor dan *use case*, maka *use case* diagram dapat digambarkan. Aktor yang berinteraksi dengan system ini adalah *user* yang terdiri atas satu jenis yaitu : *user*. Sistem dapat melakukan penyisipan dan ekstraksi seperti pada gambar 1.



Gambar 1 *Use Case Diagram* yang akan dikembangkan.

Use Case Diagram seperti yang terlihat pada gambar 1, dapat dilihat bahwa pengguna berinteraksi dengan aplikasi melalui tiga *use case* yang berbeda yaitu *use case* Input Teks Rahasia, *use case* Input Citra, *use case Steganografi Diversity* . Adapun penjelasan dari tiap *use case* adalah sebagai berikut:

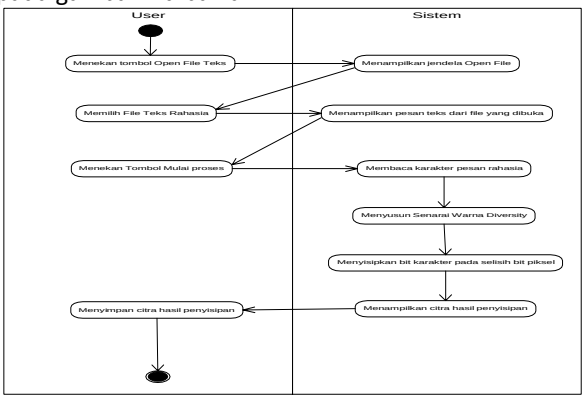
1. *Use Case Input Teks Rahasia*, *use case* ini mendeskripsikan bagaimana pengguna berinteraksi dengan melakukan input teks rahasia yang akan disisipkan kedalam citra penampung. Pengguna memilih menu yang tersedia untuk membuka file teks atau dengan mengisi langsung teks rahasia pada kolom yang disediakan.
2. *Use Case Input Citra*, *use case* ini mendeskripsikan bagaimana pengguna berinteraksi dengan melakukan input file citra digital yang akan disisipkan dengan pesan rahasia. Pengguna memilih menu yang tersedia untuk membuka file citra digital yang akan digunakan.
3. *Use Case Steganografi Diversity* , *use case* ini mendeskripsikan bagaimana pengguna melakukan proses penyisipan dan ekstraksi terhadap citra penampung. Pengguna dapat menekan menu atau tombol yang telah disediakan untuk memulai proses penyisipan maupun ekstraks pesan rahasia dari file citra penampung.

3.4 Pemodelan Dengan Menggunakan *Activity* Diagram

Acitivity Diagram merupakan diagram yang memperlihatkan urutan – urutan aktifitas yang dilakukan pengguna selama menggunakan sistem serta proses dan respon sistem yang dilakukan atas interaksi pengguna. Pada tugas akhir ini terdapat dua activity diagram utama, yaitu aktifitas diagram penyisipan dan aktifitas diagram ekstraksi.

3.5 *Activity* Diagram Penyisipan

Activity Diagram penyisipan digunakan untuk memperlihatkan alur aktivitas pengguna dalam menjalankan proses penyisipan pada aplikasi yang dibangun. Aktifitas diagram menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukan langkah-langkah dalam proses-proses kerja sistem yang kita buat di dalam sistem. Adapun langkah-langkah diagram dari *Activity Diagram* penyisipan dapat dilihat pada gambar 2 dibawah ini.

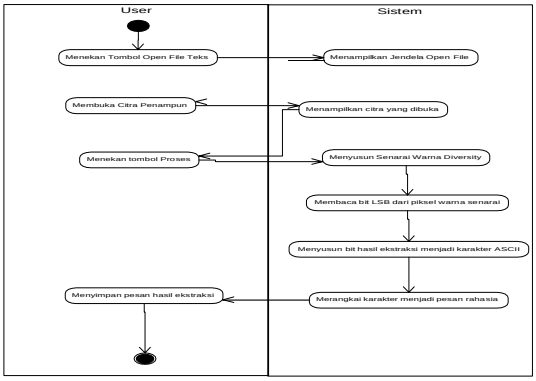


Gambar 2 *Activity Diagram* Proses *Penyisipan.*

Activity diagram proses penyisipan terlihat pada gambar 2, bahwa pengguna (*user*) berinteraksi dengan sistem aplikasi. Dimana proses langkah awal pengguna menekan tombol *open file* teks, sampai ketujuan akhir yaitu menyimpan citra hasil penyisipan data, dimana data yang disisipkan pada selisih bit piksel dapat tersimpan kedalam citra, dengan menyusun senarai warna *diversity*.

3.6 Activity Diagram Ekstraksi

Activity Diagram ekstraksi digunakan untuk memperlihatkan alur aktivitas pengguna dalam menjalankan proses ektraksi pada aplikasi yang dibangun. Adapaun diagram dari *Activity Diagram Ekstraksi* dapat dilihat pada gambar 3 dibawah ini.



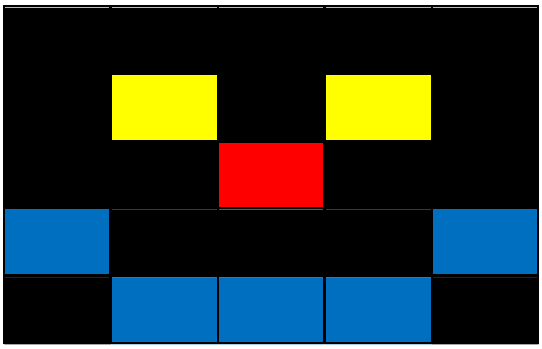
Gambar 3 *Activity Diagram* Proses Ekstraksi.

Activity diagram proses ekstraksi terlihat pada gambar 3, bahwa pengguna (*user*) berinteraksi dengan sistem aplikasi. Dimana proses langkah awal pengguna menekan tombol *open file* teks, sampai ketujuan akhir yaitu menyimpan pesan hasil ekstraksi, dimana ekstraksi dapat menyusun senarai warna *diversity* dan dapat juga membaca senarai warna piksel yang disisipkan data pada citra digital.

3.7 Analisis Metode Steganografi *Diversity*

Steganografi menggunakan metode algoritma *diversity* pada konsep nya adalah menyisipkan bit karakter pada senarai warna terpilih yang memiliki jarak warna yang berjauhan. Adapun langkah – langkah proses steganografi pada metode *diversity* adalah sebagai berikut :

1. Menyiapkan citra digital yang akan disisipkan :



Gambar 4 Citra Digital.

2. Menyiapkan pesan yang akan disembunyikan : UNHAR dan mengkonversi karakter ke dalam bit biner.
   U = 01010101, N = 01001110, H = 01001000, A = 01000001, R = 01010010
   Deretan bit :
   01010101010011100100100001000001010100100
3. Menyusun histogram warna dari citra yang digunakan.

Tabel 1 Menyusun Histogram Warna.

No	Warna	Frekuensi	Posisi Piksel
1	0,0,0 (Hitam)	17	P1, P2, P3, P4, P5, P6, P8, P10, P11, P12, P14, P15, P17, P18, P19, P21, P25
2	255, 255, 0 (Kuning)	2	P7, P9
3	255,0,0 (Merah)	1	P13
4	0,0,255 (Biru)	5	P16, P20, P22, P23, P24

4. Mengambil piksel dengan frekuensi terbesar untuk masuk ke dalam senarai warna awal. Piksel terpilih adalah piksel hitam (0,0,0) dengan frekuensi 17 piksel.
5. Berikutnya adalah menghitung jarak warna terjauh dengan warna acuan yang sudah berada di dalam senarai warna awal.
   $$D(Kuning, Hitam) = \sqrt{(255-0)^2+(255-0)^2+(0-0)^2} = 360.62$$
   $$D(Merah, Hitam) = \sqrt{(255-0)^2+(0-0)^2+(0-0)^2} = 255$$
   $$D(Biru, Hitam) = \sqrt{(255-0)^2+(255-0)^2+(0-0)^2} = 255$$
6. Dari perhitungan jarak diatas diperoleh warna terjauh dari warna senarai awal adalah warna kuning sehingga warna tersebut akan dimasukkan ke dalam senarai warna dimana senarai warna menjadi :
   Senarai Warna = {Hitam, Kuning}
7. Proses dilanjutkan dengan mengulangi langkah 5 namun menggunakan dua acuan warna yaitu hitam dan kuning.
   $$D(Merah, Hitam) = \sqrt{(255-0)^2+(0-0)^2+(0-0)^2} = 255$$
   $$D(Merah, Kuning) = \sqrt{(255-255)^2+(0-255)^2+(0-0)^2} = 255$$
   $$D(Biru, Hitam) = \sqrt{(0-0)^2+(0-0)^2+(255-0)^2} = 255$$

$$D(Biru, Kuning) = \sqrt{(0-255)^2 + (0-255)^2 + (255-0)^2} = 441.67$$

8. Dari perhitungan jarak diatas maka dapat diperoleh Biru adalah warna terjauh dari senarai warna saat ini sehingga Biru akan dimasukkan ke dalam senarai warna.
   Senarai Warna = {Hitam, Kuning, Biru}
9. Dikarenakan warna yang tersisa adalah Merah maka Merah juga akan dimasukkan kedalam senarai warna pada posisi terakhir.
   Senarai Warna = {Hitam, Kuning, Biru, Merah}
10. Proses penyisipan dilakukan pada piksel dari senarai warna yang pertama yaitu senarai warna frekuensi tertinggi adalah warna hitam dan diteruskan pada senarai warna berikutnya yaitu kuning, biru dan merah jika senarai pertama tidak mencukupi.
11. Pada senarai warna diperoleh warna pertama adalah Hitam yang mana terdiri dari 17 buah piksel yang akan digunakan sebagai piksel penampung sebagai berikut.
    P1 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikut awal dari pesan yaitu 010 pada posisi LSB sehingga warna baru dari P1 adalah :
    P1 = 0000 0000, 0000 0001, 0000 0000 = 0,1,0
    Deretan bit sisa : 10101010011100100100001000000101010010
    P2 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 101 pada posisi LSB sehingga warna baru dari P2 adalah :
    P2 = 0000 0001, 0000 0000, 0000 0001 = 1,0,1
    Deretan bit sisa : 01010011100100100001000000101010010
    P3 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 010 pada posisi LSB sehingga warna baru dari P3 adalah :
    P3 = 0000 0000, 0000 0001, 0000 0000 = 0,1,0
    Deretan bit sisa : 10011100100100001000000101010010
    P4 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 100 pada posisi LSB sehingga warna baru dari P4 adalah :
    P4 = 0000 0001, 0000 0000, 0000 0000 = 1,0,0
    Deretan bit sisa : 11100100100001000000101010010
    P5 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 111 pada posisi LSB sehingga warna baru dari P5 adalah :
    P5 = 0000 0001, 0000 0001, 0000 0001 = 1,1,1
    Deretan bit sisa : 00100100001000000101010010
    P6 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 001 pada posisi LSB sehingga warna baru dari P6 adalah :
    P6 = 0000 0000, 0000 0000, 0000 0001 = 0,0,1
    Deretan bit sisa : 001000010000000101010010
    P8 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 001 pada posisi LSB sehingga warna baru dari P8 adalah :
    P8 = 0000 0000, 0000 0000, 0000 0001 = 0,0,1
    Deretan bit sisa : 0000100000101010010
    P10 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 000 pada posisi LSB sehingga warna baru dari P10 adalah :
    P10 = 0000 0000, 0000 0000, 0000 0000 = 0,0,0
    Deretan bit sisa : 0100000101010010
    P11 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 010 pada posisi LSB sehingga warna baru dari P11 adalah :
    P11 = 0000 0000, 0000 0001, 0000 0000 = 0,1,0
    Deretan bit sisa : 0000101010010
    P12 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 000 pada posisi LSB sehingga warna baru dari P12 adalah :
    P12 = 0000 0000, 0000 0000, 0000 0000 = 0,0,0
    Deretan bit sisa : 0101010010
    P14 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 010 pada posisi LSB sehingga warna baru dari P14 adalah :
    P14 = 0000 0000, 0000 0001, 0000 0000 = 0,1,0
    Deretan bit sisa : 1010010
    P15 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari

Volume 1 Nomor 1, April 2018
e-ISSN : 2621-234X

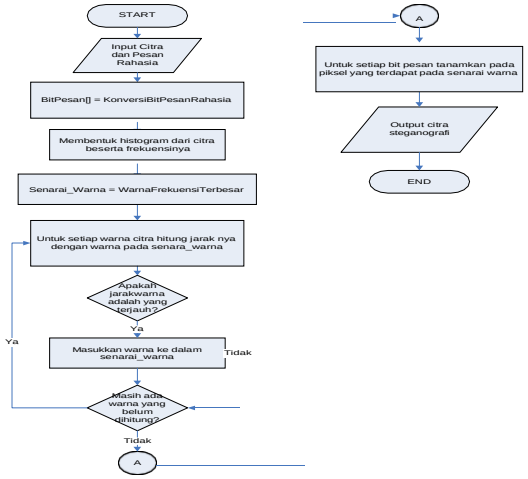
pesan yaitu 010 pada posisi LSB sehingga warna baru dari P15 adalah :
P15 = 0000 0001, 0000 0000, 0000 0001 = 1,0,1
Deretan bit sisa : 0010
P17 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 001 pada posisi LSB sehingga warna baru dari P17 adalah :
P17 = 0000 0000, 0000 0000, 0000 0001 = 0,0,1
Deretan bit sisa : 0
P18 = 0,0,0 = 0000 0000, 0000 0000, 0000 0000 disisipkan dengan tiga buah bit berikutnya dari pesan yaitu 000 pada posisi LSB sehingga warna baru dari P18 adalah :
P18 = 0000 0000, 0000 0000, 0000 0000 = 0,0,0
Deretan bit sisa :_

12. Setelah proses penyisipan selesai tahap berikutnya adalah menulis kembali nilai warna hasil penyisipan ke dalam citra yang digunakan sebagai piksel penampung pada senarai warna pertama, yaitu warna hitam yang terdiri dari 17 piksel. Terlihat pada tabel 2.

Tabel 2 Hasil Penyisipan kedalam Citra.

No	Piksel	Warna Lama	Warna Baru
1	P1	0,0,0	0,1,0
2	P2	0,0,0	1,0,1
3	P3	0,0,0	0,1,0
4	P4	0,0,0	1,0,0
5	P5	0,0,0	1,1,1
6	P6	0,0,0	0,0,1
7	P8	0,0,0	0,0,1
8	P10	0,0,0	0,0,0
9	P11	0,0,0	0,1,0
10	P12	0,0,0	0,0,0
11	P14	0,0,0	0,1,0
12	P15	0,0,0	1,0,1
13	P17	0,0,0	0,0,1
14	P18	0,0,0	0,0,0

Hasil penyisipan piksel kedalam citra dari warna lama ke warna baru dari perhitungan warna pertama adalah warna hitam yang terdiri dari 17 piksel sebagai piksel penampung, sehingga dengan menggunakan analisis algoritma di atas maka dapat diambil alur algoritma dan dimasukkan ke dalam *flowchart* seperti gambar 5 dibawah ini:



Gambar 5 *Flowchart Algoritma Diversity*

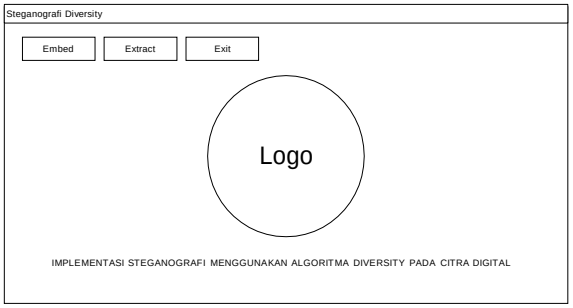
Flowchart algoritma diversity terlihat pada gambar 5, bahwa susunan alur algoritma dari mulai tombol start sampai selesai, yaitu alur proses langkah-langkah hasil penyisipan piksel ke dalam citra dari warna lama ke warna baru sebagai piksel penampung, sehingga menggunakan analisis algoritma dapat diambil alur algoritma dan dimasukan kedalam *flowchart* tersebut.

3.8 Perancangan Sistem

Sistem akan dibangun menggunakan bahasa pemrograman Visual Basic dengan menggunakan *software Microsoft Visual Studio 2010* . Rancangan antar muka disesuaikan dengan kebutuhan dan *software* yang digunakan. Antar muka menggunakan 3 *form*, yaitu terdiri dari *form cover, form* penyisipan, dan *form ekstraksi*, seperti tampilan sebagai berikut :

1. Antarmuka *Form Cover*

Pada *form cover* terdapat menu utama dimana terdiri dari tiga menu diantaranya menu *embed,* menu *extract* dan menu *exit,* terlihat pada gambar 6.



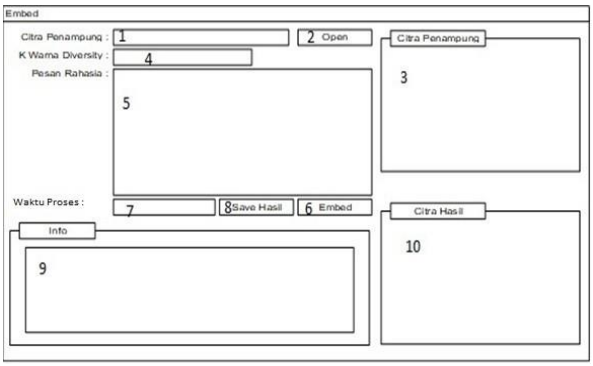
Gambar 6 Rancangan *Form Cover.*

Pada gambar 6 komponen yang dipakai untuk membangun antar muka *form Cover* adalah sebagai berikut :

a. Menu "*Embed*" : untuk masuk ketampilan Penyisipan.
b. Menu "*Extract*" : untuk masuk ketampilan Ekstraksi.
c. Menu "*Exit*" : untuk keluar dari program.

2. Antarmuka *Form Penyisipan*

Pada *form penyisipan* terdapat antarmuka yang bertujuan untuk melakukan penyisipan pesan teks menggunakan metode algoritma *diversity* menggunakan citra tampung yang dipilih oleh pengguna. Proses penyisipan menggunakan jumlah K bit yang di-input oleh pengguna untuk menentukan bit proses substitusi. Terlihat pada gambar 7.



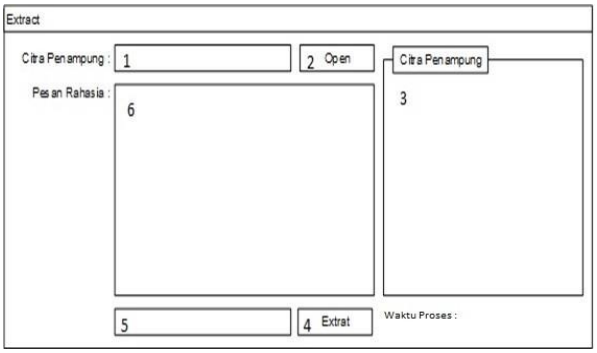
Gambar 7 Rancangan *Form* Penyisipan.

Pada gambar 7 halaman ini akan digunakan untuk memasukkan pesan rahasia yang diinginkan kedalam citra stegano untuk proses penyisipan pesan rahasia. Pada halaman ini juga pengguna mengetikkan atau menentukan sendiri pesan rahasia yang akan dimasukkan kedalam citra menggunakan algoritma steganografi *diversity*. Adapun penjelasan dari rancangan *form* penyisipan yang terlihat dari gambar 7, yaitu:

a. No.1, yang terlihat pada gambar 3.7 dimana akan menampilkan nama dari format citra penampung tersebut.
b. No.2, tombol *open*, dimana tombol tersebut dikhususkan untuk membuka atau memasukan *file* citra penampung dengan format .jpg yang ingin kita masukan pesan rahasianya, melalui proses *embed*.
c. No.3, dimana posisi letak citra penampung yang telah di masukan melalui proses No.2.
d. No.4, dimana pengguna akan memasikan K warna *Diversity* sesuai dengan yang dibutuhkan citra penampung, untuk menentukan sebagai nilai jarak atau perbedaan antar piksel.
e. No.5, dikhususkan untuk memasukkan pesan rahasia atau memasukan data teks.
f. No.6, tombol *embed* adalah tombol penyisipan pesan rahasia kedalam citra penampung.
g. No.7, dimana proses dari *embed* atau penyisipan data kedalam citra penampung.
h. No.8, tombol save hasil dari proses penyisipan berhasil.
i. No.9, dimana informasi ukuran dari data yang di embed atau di sisipkan kedalam citra penampung.
j. No.10, yaitu citra hasil atau dari citra penyelesaian proses penyisipan data rahasia kedalam citra penampung.

3. Antarmuka *Form* Ekstraksi

Pada *form extract* akan digunakan saat melakukan ekstraksi data citra yang rahasia, dimana data yang telah disisipkan oleh proses steganografi dengan metode diversity, terlihat pada gambar 8.



Gambar 8 Rancangan *Form* Ekstraksi.

Pada gambar 8 halaman ini dikhususkan untuk melakukan proses ekstraksi data pesan rahasia, yang sebelumnya sudah di input ke dalam *form* penyisipan atau disisip kedalam citra steganografi, data pesan rahasia yang telah disisipkan akan di ekstraksi kedalam

form ekstraksi tersebut. Adapun penjelasan dari rancangan *form* ekstraksi yang terlihat dari gambar 8, yaitu:

a. No.1, yang terlihat pada gambar 3.8 dimana akan menampilkan nama dari format citra penampung yang telah melalui proses penyisipan data rahasia.

b. No.2, tombol *open*, dimana tombol tersebut dikhususkan untuk membuka atau memasukan *file* citra penampung dengan format .bmp yang sudah disisipkan data melalui proses *form* penyisipan, yang akan di ekstraksi data pesan rahasianya.

c. No.3, dimana posisi letak citra penampung yang telah di masukan melalui proses No.2

d. .No.4, tombol *extract* dikhususkan untuk melakukan proses ekstraksi data pesan rahasia, yang sebelumnya sudah diinput kedalam *form* penyisipan.

e. No.5, proses dari *extract* data rahasia.No.6, tampilan data pesan rahasia yang sudah di ekstraksi.

4. Implementasi Sistem

Implementasi sistem merupakan tahap bagaimana sistem untuk dijalankan berdasarkan desain yang telah dibuat dan dirancang pada tahap sebelumnya, kemudian dimasukkan ke bahasa pemograman yang digunakan ke dalam sistem. Implementasi ini menggunakan bahasa pemograman VB.NET.

4.1 Implementasi Metode Diversity

Penerapan metode steganografi *diversity* pada sistem yang dibangun yaitu dalam proses penyisipan data teks pada data citra digital dengan ekstensi .bmp. Pengujian ini akan menggunakan citra dengan bentuk seperti yang terdapat pada gambar 9.

Gambar 9 Citra .bmp Uji Coba.

Pada gambar 4.1 citra digital di atas akan dijadikan sebagai citra penampung yang akan bekerja sebagai penampung informasi rahasia yang akan melewati proses steganografi. Selanjutnya setelah menentukan citra uji atau citra penampung, penulis akan membuka aplikasi steganografi *diversity* yang telah dibuat. Sebelum hal ini dilakukan penulis akan terlebih dahulu menjabarkan halaman-halaman yang ada beserta dengan fungsi dari masing-masing halaman yang ada pada aplikasi.

4.2 Antarmuka Sistem

Pada sistem steganografi menggunakan algoritma *diversity* yang telah dibuat, terdapat beberapa form yaitu, *Form* Menu Utama, *Form Embed*, *Form Extract.*

4.3 *Form* Menu Utama

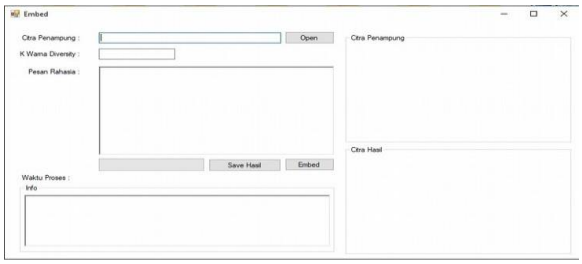
Form ini merupakan yang pertama muncul ketika membuka sistem. Pada *form* ini akan diminta *user* untuk memilih antara proses *embed* data , *extract* data , dan *exit*, terlihat pada gambar 10.

Gambar 10 *Form* Menu Utama.

Pada gambar 10 *form* ini merupakan yang pertama muncul ketika membuka sistem. Pada *form* ini akan diminta *user* untuk memilih antara proses *embed* data yang akan sisipkan teks atau pesan rahasia , *extract* data yaitu untuk membuka pesan rahasia yang telah disisipkan, dan *exit* adalah tombol keluar dari *form* utama

4.4 Form Embed

Seperti yang terlihat pada gambar 10 yang merupakan *form* utama. Setelah menekan tombol *embed*, maka sistem akan menampilkan halaman seperti pada gambar 11.

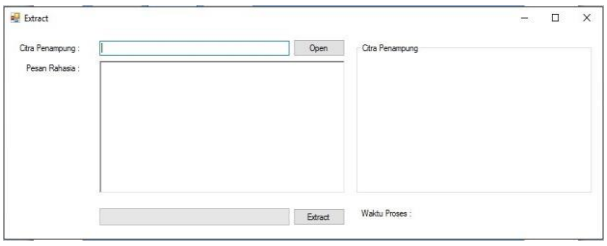


Gambar 11 *Form Embed.*

Pada gambar 11 Pada *form embed, user* diminta untuk memasukan citra uji coba kedalam citra penampung, *user* juga akan memasukan K warna diversity dengan jumlah yang diinginkan, untuk pesan rahasia akan dimasukan dan akan dilakukan proses *embed* yang nantinya hasil *embed* akan di save terlebih dahulu sebelum di ekstraksi.

4.5 Form Extract

Pada *form extract* ini akan ditampilkan halaman yang dikhususkan untuk proses ekstraksi data dari citra yang sudah melewati proses steganografi , terdapat pada gambar 12.



Gambar 12 *Form Extract.*

Pada gambar 12 Pada *form extract, user* diminta untuk memasukan citra uji coba yang telah melewati proses stegano atau citra yang telah di *embed* kedalam citra penampung*, user* akan melakukan *extract* data untuk menampilkan pesan rahasia pada citra digital.

4.6 Pengujian Sistem

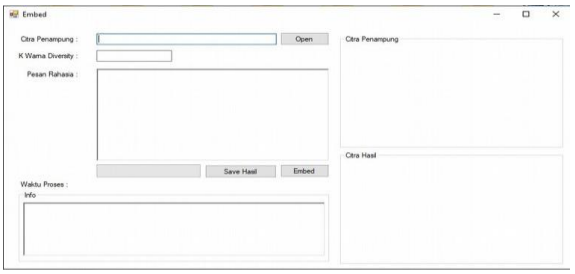
Pengujian sistem dilakukan untuk mengetahui bagaimana jalannya kerja sistem dalam melakukan steganografi data citra digital .bmp atau embed dan kemudian mengembalikannya ke kondisi semula dengan proses extract atau mengambil kembali informasi dari citra digital.

4.7 Pengujian Embed Pesan Pada Citra Digital Pada

bagian ini, dimana akan dilakukan pengujian fungsi kompresi dari algoritma yang digunakan dalam sistem. Pengujian dilakukan menggunakan data citra digital dengan proses steganografi pada algoritma *diversity,* dengan format.jpg yang akan menjadi fotmat .bmp setelah proses dilakukan, dan citra yang akan dimasukan dengan nama Ibu dan Anak.jpg dengan besar data 57 KB. Terlihat pada gambar 13.

Gambar 13 Citra uji coba.

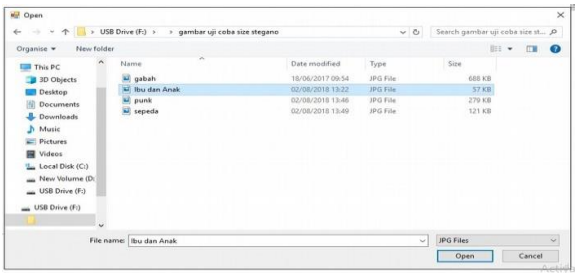
Pada gambar 13 selanjutnya pengguna akan membuka aplikasi dan memilih tombol *embed*. Setelah menekan tombol *embed* selanjutnya pengguna akan dipindahkan pada halaman khusus untuk proses steganografi. Berikut tampilan halaman kompresi sistem pada gambar 14.



Gambar 14 Halaman *Embed*.

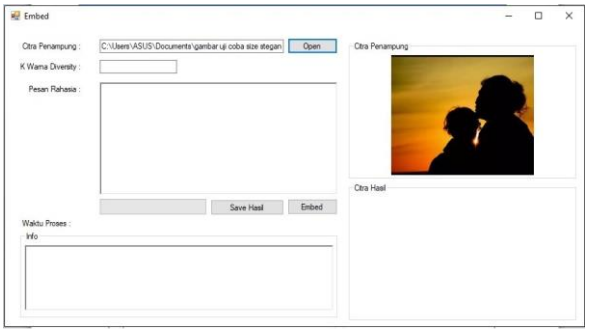
Pada gambar 14 selanjutnya pengguna akan memasukkan gambar pada sistem. Gambar yang akan dimasukan adalah gambar yang sebelumnya sudah diperkenalkan yaitu Ibu dan Anak.jpg dengan besar data 57 KB. Berikut tampilan sistem pada saat pemilihan data citra digital yang akan diinput kedalam sistem, terlihat pada gambar 15.

Volume 1 Nomor 1, April 2018
e-ISSN : 2621-234X



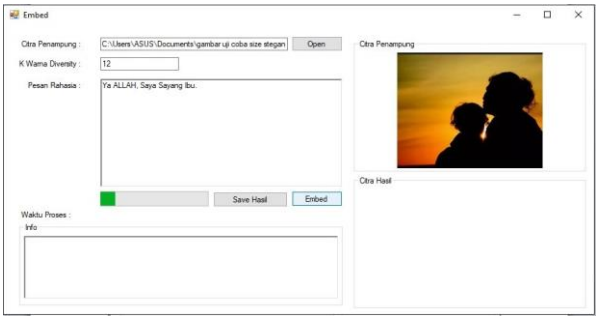
Gambar 15 Pemilihan Citra Digital Untuk Kompresi.

Pada gambar 15 setelah proses memasukan gambar citra digital sudah selesai maka halaman penyisipan kompresi akan menampilkan data pada citra penampung, terlihat pada gambar 16.



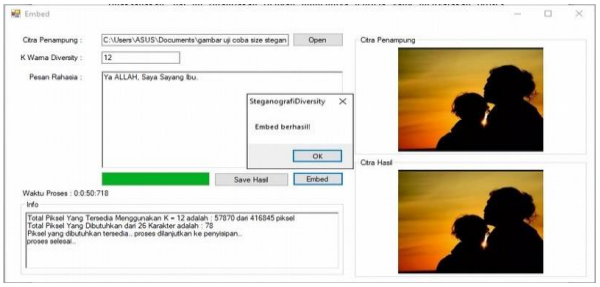
Gambar 16 *Input* Citra Kedalam Sistem.

Pada gambar 16 setelah data siap di masukan kedalam sistem yaitu citra penampung langkah selanjutnya adalah memasukkan jumlah K warna *diversity*, sabagai penentu warna *diversity,* yang kemudian akan dilanjutkan dengan memasukkan pesan rahasia yang ingin di masukkan pada citra. Pada pengujian ini pengguna akan memasukkan pesan sebagai berikut " Ya ALLAH, Saya Sayang Ibu " dengan panjang K=12. Setelah itu, penulis akan menekan tombol "*Embed*" yang akan menjalankan sistem secara otomatis dan akan menyisipkan data pesan teks pada citra digital yang di pilih, sistem akan menjalankan proses penyisipan kompresi secara otomatis seperti yang tampak pada gambar 17.



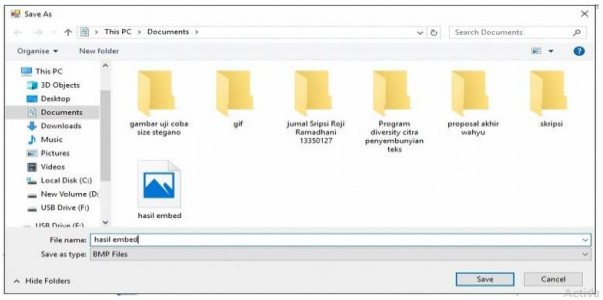
Gambar 17 Proses *Embed* Pesan Pada Citra.

Pada gambar 17 selama proses kompresi dilakukan, sistem pertama kali akan membaca data piksel dan kemudian memetakannya dan mengubahnya sesuai dengan kamus data dari algoritma. Setelah beberapa waktu, proses *embed* pun selesai dilaksanakan, hal ini ditandakan dengan munculnya jendela yang menyatakan proses *embed* berhasil, terlihat pada gambar 18.



Gambar 18 Hasil kompresi TCC.

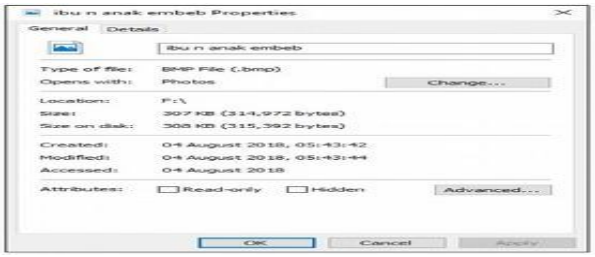
Pada gambar 18 setelah proses embed data teks berhasil dilakukan langkah selanjutnya adalah menekan tombol "OK" dan menyimpan citra yang sudah melewati proses steganografi, yang nantinya dilakukan pengujian *extract* data dari citra stegano. Format *file* akan berubah menjadi format .bmp, terlihat pada gambar 19.



Gambar 19 Penyimpanan Hasil Steganografi.

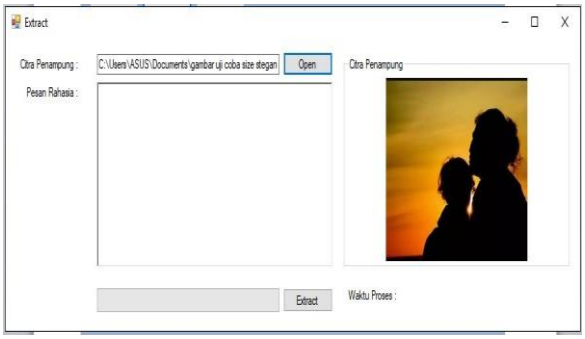
Volume 1 Nomor 1, April 2018
e-ISSN : 2621-234X

Pada gambar 19 setelah disimpan, file akan mengalami perubahan yang signifikan yang diakibatkan oleh perubahan extensi data dan juga dikarenakan adanya tambahan data pada piksel yang ada didalam citra, terlihat pada gambar 20.



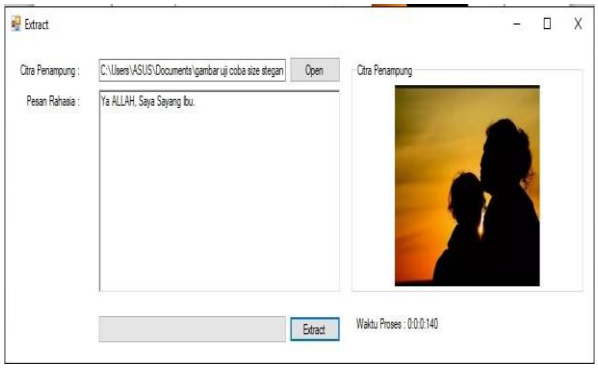
Gambar 20 Data Hasil Steganografi.

Pada gambar 20. selanjutnya memasukan citra dengan menggunakan data citra digital yang sama atau hasil steganografi, pengguna akan melanjutkan pengujian dengan melakukan proses ekstraksi data dari citra steganografi. Proses ekstraksi dapat dilihat seperti yang tampak pada gambar 21.



Gambar 21 Ekstraksi Data Dari Citra Stegano.

Pada gambar 21 setelah proses pemilihan gambar stegano selesai, selanjutnya pengguna akan menekan tombol *extract* yang ada didalam sistem seperti yang tampak pada gambar 22.



Gambar 22 Proses Kompresi Selesai.

Pada gambar 22 setelah proses *extract* selesai dapat dilihat bahwa data atau teks rahasia yang sebelumnya kita masukkan kedalam citra dapat di *extract* dengan baik dan menghasilkan informasi yang sama tanpa mengurangi data asli dari informasi itu sendiri.

Berdasarkan hasil pengujian yang dilakukan oleh penulis dalam proses *embed* dan *extract* data pada citra digital, dapat diambil kesimpulan bahwa sistem telah menerima implementasi algoritma *diversity* dengan baik dan mampu melaksanakan proses steganografi dengan baik dan sesuai dengan algoritma yang diterapkan. Berikut adalah tabel 3, data hasil size citra awal dan data hasil size citra yang telah di proses steganografi.

Tabel 3 hasil size citra awal dan size stegano.

Citra Digital	Size Awal	Size Akhir	Waktu Embed	Waktu Extract
	Type: File (.jpg). . . . Size: 57 KB. Size: 60.0 KB.	Type: File (.bmp). . . . Size: 307 KB. Size: 308 KB.	0:0:50:718	0:0:0:140
	Type: File (.jpg). . . . Size: 120 KB Size: 124 KB	Type: File (.bmp). . . . Size: 471 KB Size: 472 KB	0:0:5:265	0:0:0:132
	Type: File (.jpg). . . . Size: 278 KB Size: 280 KB	Type: File (.bmp). . . . Size: 1.37 MB Size: 1.37 MB	0:3:20:156	0:0:0:175

Dari keseluruhan piksel yang dimiliki oleh citra digital, piksel yang digunakan untuk *diversity*

kemungkinan lebih kecil dari total piksel tersebut. Dengan menentukan nilai K pada sistem algoritma *diversity,* sistem akan mencari piksel-piksel yang memiliki nilai K tersebut. Namun ukuran citra secara keseluruhan akan berubah pada saat proses steganografi selesai, yaitu *size* awal dan *size* setelah steganografi akan menjadi lebih besar.

5. Kesimpulan

Berdasarkan penelitian pengembangan aplikasi steganografi dengan menggunakan algoritma Diversity maka dapat diambil beberapa kesimpulan yaitu:

1. Implementasi steganografi menggunakan algoritma *Diversity* dalam penyembunyian data pada citra digital sudah cukup baik.
2. Dalam merancang program aplikasi pengamanan data pada citra digital menggunakan metode algoritma *Diversity* sudah cukup baik.
3. Penerapan metode algoritma *Diversity* dalam aplikasi pengamanan data pada citra digital dengan konsep steganografi sudah cukup baik.

5.1 Saran

Saran-saran yang penulis kemukakan diharapkan dapat lebih meningkatkan hasil yang telah didapatkan. Berikut ini beberapa saran yang disampaikan oleh penulis adalah:

1. Diharapkan pada penelitian yang akan datang dapat dilakukan pengujian perbandingan daya tahan dengan metode steganografi lainnya.
2. Media rahasia masih menggunakan teks standart diharapkan dapat dikembangkan menggunakan media lainnya seperti suara, video atau media lainnya.
3. Pada penulisan Tugas Akhir ini hanya menggunakan metode algoritma diversity dalam penyembunyian pesan rahasia, untuk selanjutnya bisa dikembangkan pada metode yang lebih baik.

DAFTAR PUSTAKA

[1] S. E. E. Profile, “Analisa Perbandingan Least Significant Bit Dan End Of File,” No. July, 2017.

[2] T. B. Harjo, M. Kapriati, And D. A. Susanto, “Aplikasi Steganografi Menggunakan Lsb (Least Significant Bit) Dan Enkripsi Triple Des Menggunakan Bahasa Pemrograman C #,” *Sifotek Glob.*, Vol. 6, No. 1, Pp. 13–17, 2016.

[3] R. Nova And S. Rangkuti, “Analisis Dan Perancangan Pengamanan Data Pada Citra Digital Dengan Algoritma Least Significant Bit (Lsb),” Pp. 86–97, 2014.

[4] D. Apriliani And Murinto, “Analisis Perbandingan Teknik Segmentasi Citra Digital Menggunakan Metode Levle-Set Chan & Vese Dan Lankton,” *J. Sarj. Tek. Inform.*, Vol. 1, Pp. 232–240, 2013.

[6] M. Abdul, R. Hi, And W. A. V Files, “Implementasi Mekanisme Keamanan Data Dalam Bentuk Steganografi Dengan Metode Least Significant Bit (Lsb) Pada File Audio Wav,” *Juristek*, Vol. 5, No. 2, Pp. 202–211, 2017.