IMPLEMENTASI TEKNIK STEGANOGRAFI *LEAST SIGNIFICANT BIT (LSB)* DAN KOMPRESI UNTUK PENGAMANAN DATA PENGIRIMAN SURAT ELEKTRONIK

Dedi Darwis

Manajemen Informatika, AMIK Teknokrat
Jl. Zainal Abidin Pagar Alam, No. 9-11 Kedaton Bandarlampung, Indonesia
Email: darwisdedi@teknokrat.ac.id

Abstrak *– Penelitian ini dilakukan atas dasar perlunya keamanan data pada media digital berupa metode pengolahan data yang dapat membantu mengamankan data yang bersifat rahasia, sehingga data rahasia hanya dapat dibaca oleh orang yang diinginkan dan mengantisipasi agar data tidak terbaca oleh orang yang tidak berhak. Penelitian ini menyarankan penggunaan teknik steganografi dalam keamanan data, dimana data rahasia akan disisipkan ke dalam cover image. Metode dalam penelitian ini menggunakan pendekatan dua metode yaitu metode untuk memampatkan data dan metode Least Significant Bit (LSB) sebagai metode dalam steganografi. Metode Huffman digunakan untuk memampatkan data sebelum disisipkan ke dalam cover image sehingga dapat memperkecil ukuran data yang akan disisipkan serta stego image yang dihasilkan tidak berubah secara signifikan. Hasil dari penelitian ini menghasilkan stego image yang tidak berubah secara signifikan serta proses pengambilan pesan yang relatif sangat cepat sehingga dapat menjadi alternatif untuk keamanan data yang bersifat rahasia.*

Kata kunci: *Cover Image*, *Huffman*, LSB, Steganografi, *Stego Image*.

I. PENDAHULUAN

Masalah keamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu informasi. Dengan berkembangnya teknik pengambilan informasi secara ilegal, banyak orang yang mencoba untuk mengakses informasi yang bukan haknya. Informasi vital seperti bidang pendidikan, kesehatan, bisnis dan instansi pemerintahan tidak luput dari ancaman keamanan informasi tersebut, Terlebih dengan berkembangnya media komunikasi data yang sangat popular dewasa ini seperti aplikasi *chatting*, pesan elektronik (*email)* dan aplikasi lainnya[6]. Pertukaran informasi melalui internet memang banyak keuntungan salah satunya kecepatan dalam pengirimannya, namun di lain sisi pengiriman lewat internet memiliki kekurangan yaitu kejahatan internet (*cyber crime*) seperti penyadapan, perubahan data dan lain-lain, sehingga dibutuhkan suatu teknik yang dapat melindungi informasi tersebut salah satunya adalah teknik steganografi[1].

Steganografi merupakan teknik yang bertujuan untuk memenuhi aspek kerahasiaan sebuah pesan. Steganografi yaitu penyisipan pesan tersembunyi pada *file cover* yang berfungsi sebagai media penampung sehingga tampak seperti pesan biasa, dimana pesan yang dikirim hanya dapat dibaca oleh penerima yang memiliki hak untuk mengetahui isi pesan tersebut[4]. Pada penelitian ini, digunakan suatu metode yaitu metode *Least Significant Bit* (LSB) yaitu metode yang tidak terlalu kompleks serta penyimpanan pesan pada *cover object* juga cukup besar, sehingga memungkinkan dalam melakukan penyisipan data. Dasar dari metode ini adalah bilangan berbasis biner yaitu 0 dan 1, maka proses penerapan menjadi lebih mudah. Lebih lanjut metode ini berhubungan erat dengan ukuran 1 bit, bit yang diganti hanya bit yang paling akhir, maka *stego image* atau media penampung yang dihasilkan hampir sama persis dari sebelum dilakukan steganografi sehingga tidak mengubah *cover image* secara signifikan.

Selain itu dalam melakukan steganografi, data akan dikompresi. Kompresi yaitu proses memampatkan sesuatu yang berukuran besar sehingga menjadi kecil. Dengan demikian kompresi data berarti proses untuk memampatkan data agar ukurannya menjadi lebih kecil, sehingga data yang akan disisipkan semakin kecil dan perubahan gambar yang dihasilkan hampir tidak terlihat perubahannya. Metode yang digunakan dalam melakukan kompresi data adalah metode *huffman*, dimana metode *huffman* merupakan algoritma pemampatan yang menggunakan pendekatan statistik dan pemampatan yang dilakukan cukup besar.

II. LANDASAN TEORI

A. Kompresi

Secara teknik, kompresi berarti proses memampatkan sesuatu yang berukuran besar sehingga menjadi kecil. Dengan demikian kompresi data berarti proses untuk memampatkan data agar ukurannya menjadi lebih kecil. Pemampatan ukuran besar melalui proses kompresi hanya diperlukan sewaktu berkas tersebut akan disimpan dan atau dikirim melalui media transmisi atau telekomunikasi. Apabila berkas tersebut akan ditampilkan kembali pada layar monitor, maka data yang terkompresi tersebut harus dibongkar lagi dan dikembalikan pada format semula agar dapat dibaca kembali. Proses pembongkaran berkas yang dimampatka inilah yang disebut dekompresi. Satuan yang cukup penting dalam kompresi data adalah *compression ratio* yang menggambarkan beberapa besar ukuran data setelah melewati proses kompresi dibandingkan dengan ukuran berkas asli [7].

B. Algoritma Huffman

Algoritma *huffman* adalah algoritma pemampatan yang menggunakan pendekatan statistik. Urutan langkah proses *encode* algoritma ini adalah sebagai berikut[10].

1. Urutkan nilai-nilai *grayscale* berdasarkan frekuensi kemuculannya.
2. Gabung dua buah pohon yang mempunyai frekuensi kemunculan terkecil dan urutkan kembali.
3. Ulangi langkah 2 sampai tersisa satu pohon biner.
4. Beri label pohon biner tersebut dengan cara sisi kiri pohon diberi label 0 dan sisi kanan pohon diberi label 1.
5. Telusuri pohon biner dari akar ke daun. Barisan label-label sisi dari akar ke daun adalah kode *huffman.*

Sebagai contoh, dalam kode ASCII string **"ABBABABACAACDDD"** ditulis:

A	B	B	A	B
01000001	01000010	01000010	01000001	01000010
A	B	A	C	A
01000001	01000010	01000001	01000011	01000001
A	C	D	D	D
01000001	01000011	01000100	01000100	01000100

Gambar 1. Nilai Biner Pesan

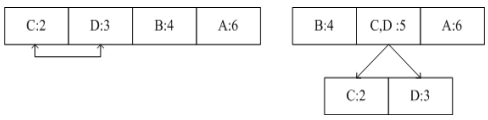
bila dikodekan menggunakan kode *huffman*, langkahnya adalah sebagai berikut:

1. Buat daftar frekuensi kemunculan tiap tiap karakter dan urutkan dari terkecil hingga terbesar.

Tabel 1**.** Tabel Frekuensi Kemunculan Karakter

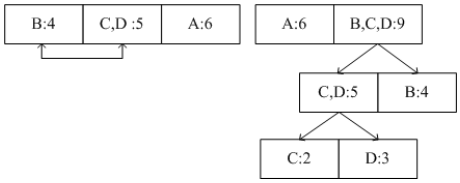
Karakter	Frekuensi
A	6
B	4
C	3
D	2

2. Gabung dua pohon yang mempunyai frekuensi kemunculan terkecil dan urutkan kembali.



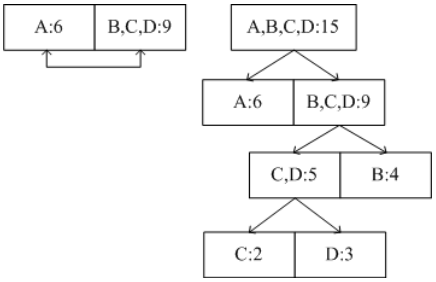
Gambar 2. Proses *Huffman* 1

3. Gabung dua pohon yang mempunyai frekuensi kemunculan terkecil dan urutkan kembali.



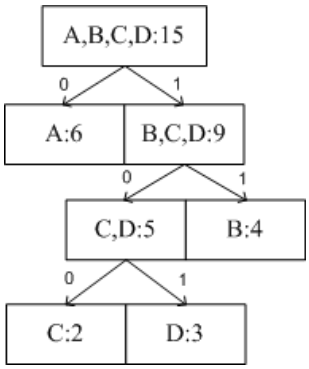
Gambar 3. Proses *Huffman* 2

4. Gabung dua buah pohon yang mempunyai frekuensi kemunculan terkecil dan urutkan kembali.



Gambar 4. Proses *Huffman* 3

5. Beri label dari akar ke daun, sebelah kiri = 0, kanan =1.



Gambar 5. Proses *Huffman* 4

Penelusuran dari akar ke daun (dari atas ke bawah) menghasilkan kode *huffman* berikut:

A=0 B=11 C=100 D=101

Tabel 2. Tabel Kode *Huffman*

Karakter	Frekuensi	Kode Huffman
A	6	0 = 1 bit
B	4	11 = 2 bit
C	2	100 = 3 bit

Dalam kode *huffman*, string "ABBABABACAACDDD" ditulis:

0 11 11 0 11 0 11 0 100 0 0 100 101 101 101 101

Dari contoh tersebut tampak bahwa kode sebuah simbol/karakter tidak boleh menjadi awalan dari kode simbol yang lain guna menghindari keraguan (*ambiguitas*) dalam proses dekompresi atau decoding. Ukuran string sebelum pemampatan (dalam kode ASCII) adalah:

=15 x 8
=120 bit

Ukuran string setelah pemampatan (dalam kode *huffman*) adalah:

=6 x1 bit + 4 x 2 bit + 3 x 3 bit + 2 x 3 bit
=29 bit

Rasio pemampatan

$$=100\% - \frac{29}{120} x\ 100\ \% = 75{,}8\ \%$$

artinya 75,8% dari string semula telah berhasil dimampatkan.

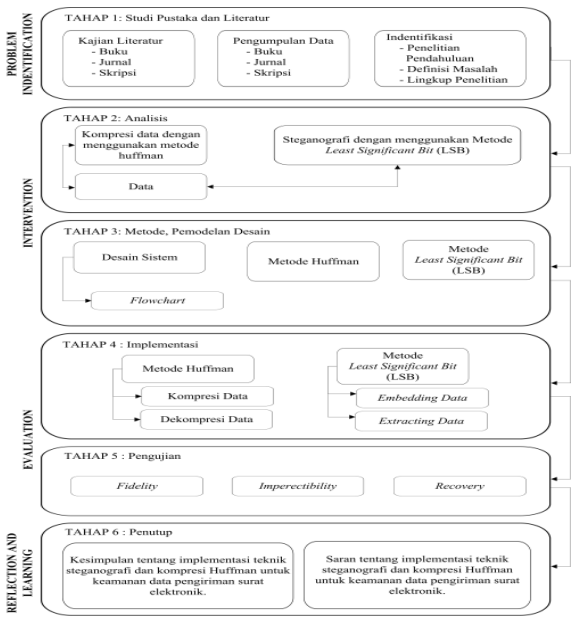
C. Pengertian Steganografi

Steganografi merupakan seni untuk menyembunyikan pesan di dalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan di dalam media tersebut. Kata steganografi (*steganography*) berasal dari bahasa Yunani *steganos* yang berarti "tersembunyi/terselubung" dan *grephien* "menulis" sehingga kurang lebih artinya "menulis (tulisan) terselubung"[10].

D. Least Significant Bit (LSB)

Metode Least Significant Bit (LSB) merupakan metode metode yang tidak terlalu kompleks, penyimpanan pesan pada cover object juga cukup besar. Dasar metode ini adalah bilangan berbasis *biner* yaitu angka 0 dan 1, karena pada data digital merupakan susunan angka 0 dan 1 maka proses penerapannya menjadi mudah. Lebih lanjut metode ini berhubungan erat dengan ukuran 1 bit dan ukuran 1 byte dimana 1 byte data terdiri dari 8 bit data dan bit pada posisi paling kanan disebut dengan LSB. Steganografi dengan metode LSB diganti dengan bit yang disembunyikan. Karena bit yang diganti hanya bit yang paling akhir, maka *stego image* yang dihasilkan hampir sama persis dengan *cover image* nya[1].

III.METODE PENELITIAN



Gambar 6. Tahapan Penelitian

A. Studi Pustaka dan Literatur

Kajian literatur mengenai penelitian ini didapat dari jurnal buku dan skripsi yang telah melakukan penelitian sebelumnya. Setelah itu dilakukan indentifikasi terhadap penelitian terdahulu, definisi masalah dan lingkup penelitian guna menunjang keberhasilan dalam penelitian.

B. Analisis

Analisis yang dilakukan dalam penelitian ini yaitu bagaimana melakukan kompresi data dengan menggunakan metode *Huffman* terhadap data yang akan di steganografi dengan menggunakan metode *Least Significant Bit* (LSB), yang bertujuan untuk menjamin kerahasiaan dan menjaga keamanan data.

C. Metode, Pemodelan Desain

Metode dan pemodelan desain yang dilakukan dalam penelitian ini yaitu desain sistem menggunakan *flowchart* sebagai gambaran alur sistem. Terdapat dua metode yang digunakan dalam penelitian ini yaitu metode *Huffman* sebagai metode dalam kompresi data dan metode *Least Significant Bit* (LSB) sebagai metode dalam melakukan steganografi.

D. Implementasi

Implementasi dalam penelitian ini yaitu membuktikan bagaimana proses kompresi menggunakan metode *Huffman* dapat melakukan kompresi dan dekompresi terhadap data rahasia, serta bagaimana proses steganografi dengan menggunakan metode *Least Significant Bit* (LSB) dapat melakukan *embedding data* dan *extracting data* terhadap data yang disisipkan.

E. Pengujian

Pengujian yang dilakukan dalam penelitian ini adalah *fidelity*, *imperectibility* dan *recovery* sebagai suatu pengujian dalam menguji keamanan dan kualitas citra data yang telah di steganografi dengan menggunakan dua metode yaitu metode *Huffman* dan metode *Least Significant Bit* (LSB).

F. Penutup

Kesimpulan dan saran dalam penelitian ini dilakukan untuk menyimpulkan penelitian yang telah dilakukan dan memberikan saran terhadap penelitian kedepannya.

IV. HASIL DAN PEMBAHASAN

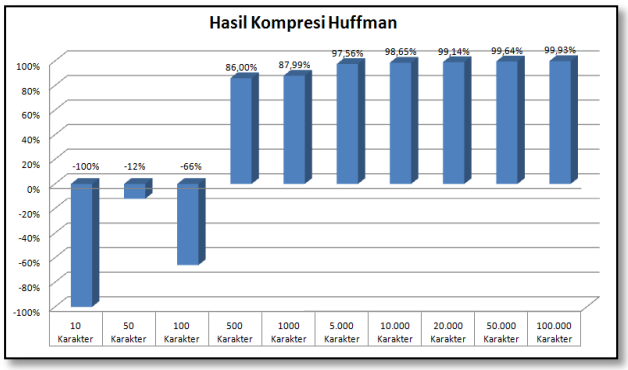
A. Hasil Pengujian Kompresi

Pengujian kompresi *Huffman* dilakukan dengan tujuan untuk mengetahui seberapa besar kompresi data yang dihasilkan. Pengujian kompresi *Huffman* ini menggunakan jumlah karakter yang disusun secara acak dan memiliki nilai kemunculan karakter yang beragam. Dari pengujian yang telah dilakukan, didapatkan hasil dari kompresi *Huffman* sebagai berikut:

Tabel 3. Tabel Pengujian Kompresi *Huffman*

No	Jumlah Karakter	Ukuran Asli (Kb)	Ukuran kompresi (Kb)	Ratio (%)
1	10 Karakter	10 byte	20 byte	-100%
2	50 Karakter	50 byte	56 byte	-12%
3	100 Karakter	100 byte	166 byte	-66%
4	500 Karakter	500 byte	70 byte	86%
5	1000 Karakter	999 byte	120 byte	87,99%
6	5000 Karakter	49997,12 byte	122 byte	97,56%
7	10.000 Karakter	10024,96 byte	135 byte	98,65%
8	20.000 Karakter	19968 byte	171 byte	99,14%
9	50.000 Karakter	49971,2 byte	179 byte	99,64%
10	100.000 Karakter	103424 byte	75 byte	99,93%

Dari perhitungan di atas didapatkan grafik hasil kompresi *Huffman* sebagai berikut:



Gambar 7. Hasil Kompresi *Huffman*

Hasil yang didapatkan dari pengujian kompresi *Huffman* di atas adalah nilai rasio yang didapatkan dari pengujian tersebut sangatlah beragam. Hal ini disebabkan dari frekuensi karakter yang di kompresi. Semakin besar frekuensi kemunculan karakter dari suatu pesan, maka nilai kompresi akan semakin besar dan jika semakin kecil frekuensi kemunculan karakter dari suatu pesan maka semakin kecil nilai rasio dari kompresi tersebut.

B. Pengujian Metode Least Significant Bit (LSB)

Pengujian metode *Least Significant Bit* (LSB) bertujuan untuk mengetahui apakah data yang telah dikompresi dapat disisipkan ke dalam *cover image*. *Cover image* yang digunakan dalam penelitian ini mewakili dari representasi warna yaitu R (*Red*), G (*Green*) dan B (*Blue*), hal ini dimaksudkan untuk menguji apakah data dapat disisipkan ke dalam *cover image* berdasarkan representasi warna yang diujikan. Penyisipan pesan dilakukan pada *cover image* dengan memperhatikan kunci yang diberikan sebelum melakukan proses *embedding* data. Kunci yang dimasukkan oleh *user* akan dibaca oleh system dan di ubah menjadi kode *biner*. Jika *biner* bernilai 0 maka data akan disisipkan pada *bit* ke enam, jika nilai *biner* bernilai 1 maka data akan disisipkan pada *bit* ke tujuh. Kode *biner* kunci akan terus diulang sampai data yang berada pada *cover image* selesai terbaca. Terdapat tiga *bit* pesan yang terdapat pada setiap piksel dari *cover image*. Berikut pengujian *embedding* data yang dilakukan.

Tabel 4. Tabel Pengujian *Embedding*

No	Nama Gambar	Ukuran Gambar	Ukuran Pesan	Status
1	Gambar 1	500 x 375	5,57 Kb	Berhasil
2	Gambar 2	720 x 480	11,2 Kb	Berhasil
3	Gambar 3	849 x 565	10,3 Kb	Berhasil
4	Gambar 4	1280 x 720	33,6 Kb	Berhasil
5	Gambar 5	1500 x 565	25,6 Kb	Berhasil

Hasil dari pengujian di atas menunjukkan bahwa penyisipan data ke dalam *cover image* berhasil dilakukan dengan baik. Terdapat kesimpulan bahwa data yang disisipkan harus sesuai dengan jumlah representasi citra dan tidak melebihi dari representasi citra, dikarenakan jumlah penampung pada *cover image* berdasarkan resolusi dari citra dari *cover image* itu sendiri.

C. Pengujian Fidelity

Pengujian *fidelity* dilakukan untuk menguji terhadap *cover image* dan *stego image*, apakah data tidak jauh berubah setelah terjadi penambahan data rahasia dan apakah *stego image* masih dapat terlihat dengan baik, sehingga pengamat tidak mengetahui bahwa di dalam *stego image* tersebut terdapat data rahasia. Pengujian *fidelity* dilakukan dengan beberapa pengujian yaitu dengan pengujian MSE (*Mean Square Error*) dan PSNR (*Peak Signal to Noise Ratio*).

1. Pengujian MSE (Mean Square Error)

Pengujian MSE (*Mean Square Error*) dilakukan untuk menentukan nilai rata – rata kuadrat dari jumlah kuadrat *absolute error* antara *cover image* dengan citra *stego*, sebelum menentukan PNSR (*Peak Signal to Noise Ratio*). Terdapat rumus dalam menghitung MSE, rumus MSE adalah sebagai berikut:

$$MSE_{AVG} = \frac{MSE_R + MSE_G + MSE_B}{3}$$

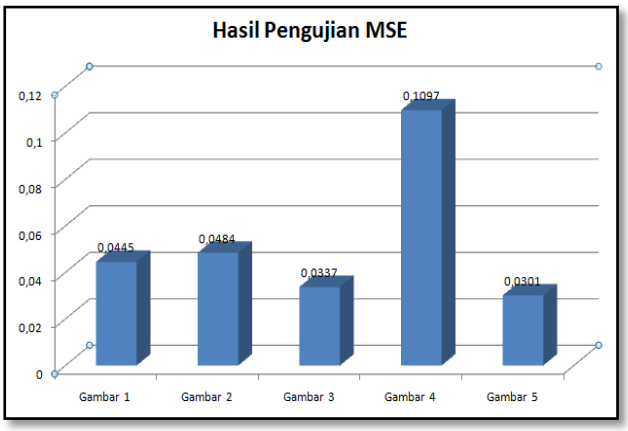
Keterangan:
MSE_{AVG} = Nilai rata-rata MSE *cover image*.
MSE_R = Nilai MSE warna merah.
MSE_G = Nilai MSE warna hijau.
MSE_B = Nilai MSE warna Biru.

Hasil perhitungan MSE yang dilakukan dalam penelitian ini, dapat dilihat pada tabel dibawah ini:

Tabel 5. Tabel Pengujian MSE (*Mean Square Error*)

No	Nama Gambar	Ukuran Gambar	Ukuran *Stego Image*	MSE
1	Gambar 1	500 x 375	228 Kb	0,0445
2	Gambar 2	720 x 480	644 Kb	0,0484
3	Gambar 3	849 x 565	284 Kb	0,0337
4	Gambar 4	1280 x 720	0,94 Mb	0,1097
5	Gambar 5	1500 x 565	1,69 Mb	0,0301

Dari pengujian yang telah dilakukan pada pengujian di atas, penulis menggunakan grafik yang bertujuan menunjukan nilai MSE yang telah dilakukan. Berikut gambar grafik nilai MSE yang dihasilkan.



Gambar 8. Grafik Hasil Pengujian MSE

Pengujian MSE yang telah dilakukan, menghasilkan nilai MSE dibawah 1 sehingga dapat dikatatakan baik.

2. Pengujian PSNR (Peak Signal to Noise Ratio)

Pengujian PNSR (*Peak Signal to Noise Ratio*) digunakan untuk mengukur kualitas citra yang dihasilkan. Metode PNSR adalah ukuran perbandingan antara nilai piksel *cover image* dengan nilai piksel pada citra *stego* yang dihasilkan. Berikut rumus, hasil perhitungan dan grafik hasil perhitungan PNSR yang telah dilakukan.

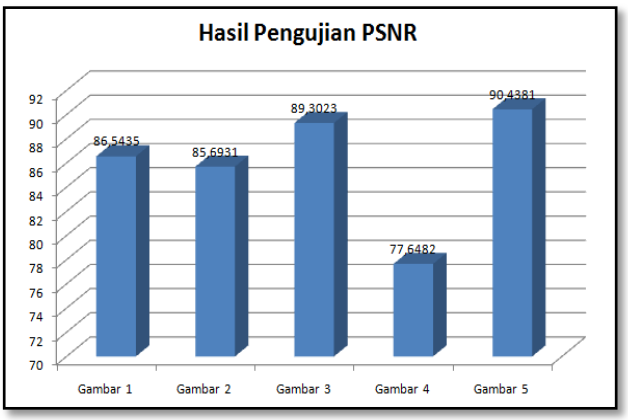
$$PSNR = 10_{log10}\left(\frac{255^2}{MSE}\right)$$

Keterangan:
PNSR = Nilai PNSR citra digital.
MSE = Nilai *Mean Square Error* dari citra.

Tabel 6. Tabel Pengujian PSNR (*Peak Signal to Noise Ratio*)

No	Nama Gambar	Ukuran Gambar	Ukuran *Stego Image*	PSNR
1	Gambar 1	500 x 375	228 Kb	86,5435
2	Gambar 2	720 x 480	644 Kb	85,6931
3	Gambar 3	849 x 565	284 Kb	89, 3023
4	Gambar 4	1280 x 720	0,94 Mb	77,6482
5	Gambar 5	1500 x 565	1,69 Mb	90,4381

Dalam memvisualisasikan data hasil pengujian PSNR, peneliti mengunakkan grafik seperti di bawah ini:

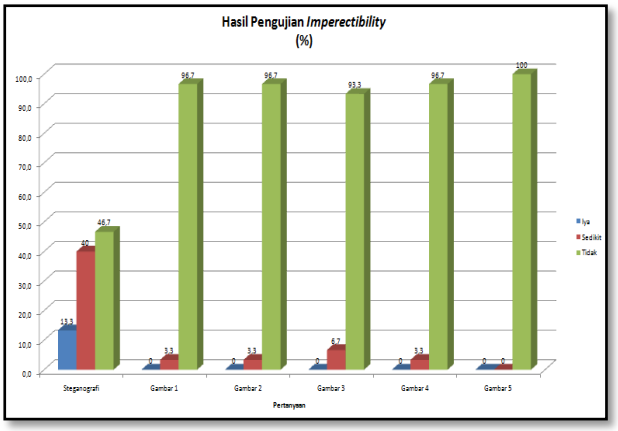


Gambar 9. Grafik Hasil Pengujian PSNR

Berdasarkan pengujian MSE dan PNSR didapatkan bahwa nilai MSE yang dihasilkan kurang dari 1 dB dan PNSR di atas 50, berarti perubahan kualitas warna antara citra asli dengan *stego image* tidak mengalami perubahan yang signifikan, sehingga keberadaan dari file yang tersembunyi tidak mudah di deteksi oleh indra penglihatan manusia.

D. Pengujian Imperectibility

Pengujian *imperectibility* dilakukan untuk menguji terhadap seberapa mudah *stego image* dapat dideteksi oleh indra manusiawi. Pengujian *imperectibility* dilakukan secara manual dengan menunjukkan secara visual *stego image* dan *cover image* terhadap beberapa responden. Hasil dari pengujian *imperectibility* yang dilakukan dari beberapa responden dapat dilihat pada grafik di bawah ini:



Gambar 10. Grafik Hasil Pengujian *Imperectibility*

Berdasarkan data-data yang didapatkan pada pengujian *imperectibility* yang telah dilakukan, menyatakan bahwa hasil steganografi yang dihasilkan terdapat banyak responden yang tidak menyadari perbedaan yang diamati. Sehingga hasil steganografi yang dihasilkan dapat disimpulkan sangat baik.

E. Pengujian Recovery

Pengujian *recovery* dilakukan untuk mengetahui apakah data yang disembunyikan pada *stego image* dapat diungkap kembali. Pengujian *recovery* digunakan untuk melihat apakah data yang telah dilakukan dapat dilepas ke bentuk semula. Proses *recovery* dimulai dengan *user* memasukkan kunci untuk melakukan *extracting data*. Proses ekstraksi diawali dengan membaca kode *biner* dari kunci yang dimasukkan. Kode biner dari kunci akan dibaca oleh sistem dan sistem akan memilah *biner* dari setiap kode *biner* yang didapati. Jika kode *biner* bernilai 0 maka sistem akan mengambil *biner* dari *cover image* dengan mengambil nilai *bit* yang ke tujuh, jika kode *biner* bernilai 1 maka sistem akan mengambil biner dari *cover image* pada bit ke delapan, proses ekstraksi berhenti ketika semua data telah didapati. Berikut hasil dari pengujian *recovery* yang telah dilakukan.

Tabel 7. Tabel Pengujian *Recovery*

No	Nama Gambar	Ukuran Gambar	Ukuran *Stego Image*	Status
1	Gambar 1	500 x 375	228 Kb	Berhasil
2	Gambar 2	720 x 480	644 Kb	Berhasil
3	Gambar 3	849 x 565	284 Kb	Berhasil
4	Gambar 4	1280 x 720	0,94 Mb	Berhasil
5	Gambar 5	1500 x 565	1,69 Mb	Berhasil

Selain pengujian di atas, peneliti melakukan pengujian terhadap gambar yang telah diubah seperti diputar dan dipotong.

Tabel 8. Tabel Pengujian *Recovery* Gambar Termodifikasi

No	Nama Gambar	Ukuran Gambar	Ukuran *Stego Image*	Perubahan	Status
1	Gambar 1	500 x 375	228 KB	Diputar	Gagal
2	Gambar 2	720 x 480	644 KB	Diputar	Gagal
3	Gambar 3	849 x 565	284 KB	Diputar	Gagal
4	Gambar 4	1280 x 720	0,94 MB	Diputar	Gagal
5	Gambar 5	1500 x 565	1,69 MB	Diputar	Gagal
6	Gambar 6	337 x 375	75,9 KB	Dipotong	Gagal
7	Gambar 7	463 x 480	110 KB	Dipotong	Gagal
8	Gambar 8	832 x 720	123 KB	Dipotong	Gagal
9	Gambar 9	849 x 382	59,2 KB	Dipotong	Gagal

Berdasarkan data-data di atas, menunjukan bahwa pengujian *recovery* yang dilakukkan tidak berhasil dikarenakan representasi pixel yang digunakan ketika melakukan *extracting data* telah diubah sebelumnya. Dalam pengujian ini dapat disimpulkan bahwa perubahan yang dilakukan pada *stego image* dapat mempengaruhi dari proses *extracting data.*

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan rumusan masalah, hasil penelitian dan pembahasan mengenai aplikasi steganografi dapat diambil beberapa kesimpulan, yaitu:

1. Steganografi dengan menggunakan metode *Least Significant Bit* (LSB) dan kompresi data dengan menggunakan metode *Huffman* dapat menyisipkan informasi ke dalam media digital.
2. Kompresi *Huffman* memiliki hasil pegujian yang beragam, disebabkan karena nilai rasio dari metode *Huffman* didasari dari frekuensi kemunculan dari karakter dari suatu pesan. Semakin besar frekuensi kemunculan dari karakter pesan, semakin besar nilai rasio kompresi yang dihasilkan dan semakin kecil frekuensi kemunculan karakter pesan semakin kecil nilai rasio kompresi yang dihasilkan.
3. Proses ekstraksi data berhasil dilakukan dengan baik. Namun, tidak dapat mengatasi perubahan *stego image* yang telah berubah.
4. Proses ekstraksi data dapat mengembalikan ukuran pesan dan isi pesan dengan baik.

B. Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut dari aplikasi steganografi yaitu:

1. Aplikasi steganografi yang dibuat belum memiliki *error detection* sehingga dibutuhkan

penelitian selanjutnya untuk dapat mengetahui apakah *stego image* mengalami perubahan oleh orang yang tidak bertanggungjawab.

2. Dalam penelitian ini, peneliti hanya menggunakan kunci yang tidak begitu baik dan rentan terhadap keamanan data. Diharapkan pada penelitian selanjutnya untuk dapat menggunakan kriptografi yang lebih baik pada pesan maupun *stego-key*.

DAFTAR PUSTAKA

[1] Aryasanti, A., & Hardjianto, M. (2014). Model Pengamanan Berkas Bank Soal Dengan Metode Steganografi LSB dan Kompresi. *Jurnal TICOM , II* (2), 127-135.

[2] Cahyono, T. D. (2008). Pemodelan Waterfall Dan Pengembangan Evolusioner Dalam Proses Rekayasa Sistem Perangkat Lunak. *J PENGEMB. REK & TEK* , 66-72.

[3] Faradisa, I. S., & Budiono, B. F. (2011). Implementasi Meode *Huffman* Sebagai Teknik Kompresi Citra. *Jurnal Elektro ELTEK , II* (2), 176-182.

[4] Hakim, Z., Permana, E. A., & Sidik, A. (2014). Analisis Dan Implementasi Teknik Steganografi Sebagai Fasilitas Pengamanan Proses Pengiriman File Secara Online. *Jurnal Sisfotek Global , I*, 18-21.

[5] Jogiyanto. (2005). *Analisis & Desain Sistem Informasi.* Yogyakarta: ANDI OFFSET.

[6] Komala, S., & Hardjianto, M. (2014). Model Keamanan Pesan Rahasia Pada Citra Menggunakan Metode One's Complement Cryptography Dan Least Significant Bit (LSB) Steganography di Perangkat Berbasis Android. *Jurnal TICOM , II* (2), 117-126.

[7] Kristanto, A. (2003). *Keamanan Data pada Jaringan Komputer.* Yogyakarta: GAVA MEDIA.

[8] Kurniawan, T. A. (2014). Pemanfaaatan Metode LSB Pada Citra Digital Dalam Mengaplikasikan Steganografi Sebagai Upaya Peningkatan Jaminan Keamanan Dalam Transaksi Informasi Secara Online. *Jurnal TICOM , II* (2), 71-79.

[9] Rosa A.S, M. S. (2013). *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek.* Bandung: Informatika.

[10] Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., & Wijanarto. (2009). *Teori Pengolahan Citra Digital.* (B.R.W,Penyunt.)Yogyakarta: ANDI.