SIMULASI DAN ANALISIS KEAMANAN TEKS MENGGUNAKAN METODE STEGANOGRAFI DISCRETE COSINE TRANSFORM (DCT) DAN METODE ENKRIPSI CELLULAR AUTOMATA

Arianto Sirandan[1], Ir. Rita Magdalena, M.T[2], Nur Andini, S.T., M.T.[3]
[1,2,3]FakultasTeknik Departemen Elektro dan Komunikasi Universitas Telkom
[1]ariantosirandan@gmail.com, [2]ritamagdalena@telkomuniversity.ac.id, [3]andini_dhine@yahoo.com

Abstrak

Keamanan suatu pesan rahasia pada saat ini menjadi hal yang sangat penting untuk ditingkatkan, mengingat begitu mudahnya siapa saja untuk mengakses ke jaringan. Oleh karena itu, keamanan pesan rahasia yang dikirimkan harus terjamin kerahasiaannya. Salah satu caranya dengan menggunakan metode steganografi. Steganografi adalah suatu teknik penyembunyian pesan rahasia ke dalam media lain tanpa diketahui orang lain. Pesan rahasia yang dikirimkan dapat berupa *text, image, voice* maupun *video.* Untuk media penyembunyian juga dapat berupa *text, image, voice* maupun *video.*

Dalam tugas akhir ini dilakukan perancangan sistem dengan menggunakan metode *Discrete Cosine Transform* dan *Cellular Automata*. Kedua metode ini digabungkan dan digunakan untuk menyembunyikan suatu pesan rahasia yang berupa *text* untuk mendapatkan tingkat keamanan yang lebih tinggi. Pada proses awal sistem, pesan disisipkan ke dalam suatu citra *cover*, setelah itu citra *cover* tersebut dienkripsi sehingga gambar asli dari citra *cover* tidak dapat diketahui.

Hasil yang diperoleh dari tugas akhir ini adalah sebuah citra CA yang berupa gambar rusak tetapi memiliki pesan rahasia di dalamnya. Dengan nilai CER tanpa *noise* pada layer *blue* yaitu 0% dan pada layer *red* dan *green* yaitu 7,98% dan 25,35%. Namun, ketika diberi *noise* pesan yang disisipkan tidak dapat diekstrak lagi. Dengan demikian, sistem ini dapat berjalan dengan baik jika tidak diberi serangan.

Kata Kunci: *Steganography, 2D Rules, Cellular Automata* (CA).

Abstract

Security of secret message is becoming an important thing to improve, remember the easiness for everyone to access the network at this time. Because of that, message security that being sent to receiver must be reliable. One of the way for improving the security is using steganography. Steganography is a technique of hiding message from one media to another without being known. A secret message can be text, image, voice or video. The media concealment can also be text, image, voice or video.

This final project is doing system design by using Discrete Cosine Transform and Cellular Automata method. Both of the methods are being combined and used to hide a secret message in the form of text for getting a higher security. In the beginning of the process, the message is being inserted into a cover image, and then the cover image will be encrypted, so the original image of cover image can not be identified.

The result of this final project is a CA image in the form of broken image but still has a secret message in it. The CER value without noise in the blue layer is 0% and in the red and green layer are 7,98% and 25,35%. But, when the image is given noise, the secret message can not be extracted anymore. It can be concluded that this system will work well if the CA image is not given attack.

***Keyword: Steganography, 2D Rules, Cellular Automata* (CA).**

1. Pendahuluan

Perkembangan teknologi saat ini sudah berkembang sangat pesat, termasuk perkembangan media komunikasi melalui internet. Namun, pertukaran informasi yang terjadi antara pengirim dan penerima sangat rentan terjadi penyadapan dan pengubahan data yang dikirimkan. Oleh karena itu, dibuat berbagai macam metode penyembunyian pesan sehingga pesan yang dikirimkan tidak mudah dibaca oleh orang lain. Steganografi adalah suatu teknik penyembunyian pesan baik *text, image, voice* ataupun *video* dengan menggunakan media lain sebagai *cover* yang juga bisa terdiri dari *text, image, voice* dan *video*.

Seiring perkembangan steganografi yang ada rupanya teknik ini tidak menjamin kerahasiaan pesan ketika dikirimkan. Pesan yang disembunyikan dengan mudah dibaca ketika *hacker/attacker* mampu membongkar algoritmanya dengan menggunakan metode steganalisis.. Sebab itu dalam Tugas Akhir ini disimulasikan proses steganografi dengan menggunakan metode *Discrete Cosine Transform* yang kemudian ditambahkan lagi metode kriptografi dengan mengenkripsi pesan yang telah disembunyikan yaitu *Cellular Automata*. Dengan cara seperti ini diharapkan tingkat keamanan dari pesan yang dikirimkan menjadi sangat tinggi dan menyulitkan para *hacker/attacker* untuk membaca pesan yang kita kirimkan.

2. Dasar Teori

2.1 Steganografi

Steganografi berasal dari bahasa Yunani yang terdiri dari kata ***steganos*** yang berarti tersembunyi dan kata ***graphein*** yang artinya menulis. Dengan kata lain, steganografi adalah tulisan tersembunyi. Stegonagrafi dapat juga diartikan sebagai suatu teknik penyembunyian pesan dengan menyisipkan pesan tersebut ke suatu media lain yang dapat berupa citra, suara, maupun teks. Steganografi memungkinkan seseorang untuk

menyisipkan suatu pesan rahasia ke dalam pesan lain tanpa diketahui oleh orang lain.

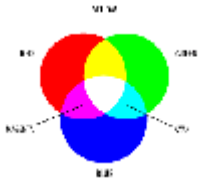
Dalam penggunaan steganografi ini, dibutuhkan 2 jenis data.Data yang pertama adalah data untuk ditumpangi oleh pesan rahasia dan data yang kedua adalah data atau pesan yang ingin disisipkan. Kedua jenis data ini dapat berupa citra, suara maupun teks.

2.2 Citra dan Citra Digital[3]

Citra merupakan suatu fungsi kontinyu dari intensitas cahaya atau derajat keabuan dalam bidang 2 dimensi yang dapat direpresentasikan dengan f(x,y), dimana x dan y merupakan koordinat spasial dan nilai f(x,y) sebanding dengan skala intensitas cahaya dari citra pada titik tersebut.

2.2.1 Citra RGB

Citra RGB (*Red Green Blue*) merupakan citra digital yang setiap pikselnya tersusun dari kombinasi tiga warna dasar yaitu merah, hijau, dan biru. Setiap warna dasar mempunyai rentang nilai dari 0 sampai 255.



Gambar 2.3 RGB

Pemilihan skala 256 ini didasarkan pada cos

penggunaan 8 digit bilangan biner dalam mesin komputer,sehingga akan diperolah warna total sebanyak 16.777.216 warna.

2.3 Format Data Citra[5]

Di dalam komputer, citra digital disimpan sebagai suatu *file* dengan format tertentu. Format citra tersebut menunjukan cara sebuah citra digital disimpan, misalnya apakah dengan suatu kompresi atau tidak. Dalam tugas akhir ini, digunakan format citra digital Bitmap sebagai citra *cover* dan citra rahasia yang disisipkan.

2.3.1 Bitmap (.bmp)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel.

2.4 *Discrete Cosine Transform* (DCT)[6]

DCT merupakan transformasi matematis yang mengambil sinyal dan mentransformasikannya dari domain spasial ke berbasis frekuensi dapat lebih cepat. Terdapat 2 jenis DCT yaitu DCT dimensi satu (DCT 1-D) dan DCT dimensi dua (DCT 2-D).

2.4.1 *Discrete Cosine Transform* 2-D(DCT 2-D)

DCT 2-D digunakan untuk mengolah sinyal-sinyal yang berdimensi dua, seperti citra yang merupakan sinyal dua dimensi. Untuk sebuah matriks yang berukuran n x m, DCT 2-D dapat dihitung dengan cara: menerapkan DCT 1-D pada setiap baris dari s dan kemudian hasilnya dihitung DCT untuk setiap kolomnya. Rumus transformasi DCT 1-D untuk s adalah sebagai berikut:

DCT dari sederet n bilangan real s(x), x=0, …, n-1

$$\text{.....(2.2)}$$

Dengan u = 0, ., n-1

Dimana $= \begin{cases} 2 \\ 1 \end{cases}$

Persamaan di atas menyatakan s sebagai kombinasi linier dari basis vektor.Koefisien adalah elemen transformasi S, yang mencerminkan banyaknya setiap frekuensi yang ada di dalam masukan s.

Sedangkan, rumus untuk transformasi DCT 2-D untuk s adalah sebagai berikut:

$=$

$$\cos \quad \cos$$

……..(2.3)

Dengan u = 0, .., n-1 ; v = 0, …, m-1

Rumus di atas sering juga disebut dengan *Forward Discrete Cosine Transform* (FDCT).DCT 2-D dapat dihitung dengan menerapkan transformasi 1-D secara terpisah pada baris dan kolomnya, sehingga kita dapat mengatakan bahwa DCT 2-D *sparable* dalam dua dimensi. Untuk *Invers Discrete Cosine Transform* dimensi dua (IDCT 2-D) dapat diperoleh dengan rumus berikut:

$=$

$$\cos \quad \cos$$

domain frekuensi. Dalam pengolahan berbasis frekuensi membutuhkan waktu yang lama namun memberikan hasil yang cukup menjanjikan, hal ini dikarenakan banyaknya jumlah frekuensi yang diamati. Hal tersebut dapat diatasi dengan perkembangan teknologi mikroprosesor yang makin canggih, sehingga proses pengolahan

……..…(2.4)

Dengan x= 0, .., n-1 ; y = 0, …,
m-1

2.5 *Cellular Automata* (CA)

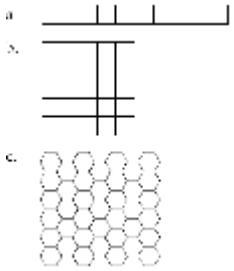
Secara teoritis, *cellular automata* pertama kali diperkenalkan pada tahun akhir tahun 1940-an oleh John Von Neumann dan Stanislaw Ulam sebagai model sederhana untuk mempelajari proses biologi seperti *self-reproduction organism.* Secara praktis, *cellular automata* berkembang ketika pada akhir tahun 1960-an John Conway membuat *game of life* yang mampu memodelkan kehidupan nyata secara sederhana.

2.5.1 Definisi *Cellular Automata*

Cellular automata adalah sebuah array dengan *automata* yang identik, atau disebut juga sel, yang saling berinteraksi satu sama lain. Array

tersebut dapat membentuk susunan sel 1 dimensi, 2 dimensi maupun 3 dimensi.

Berikut ilustrasi susunan sel-sel *cellular automata*:



Gambar 2.6 susunan sel-sel *cellular automata* Segi empat 1 dimensi, (b) segi empat 2 dimensi, (c) segi enam 2 dimensi

Unsur-unsur pembentuk *cellular automata* adalah :

1. Geometri

Geometri adalah bentuk sel serta bentuk sistem yang disusun oleh sel-sel tersebut. Geometri *cellular automata* terdiri atas dimensi *cellular automata* tersebut (1-dimensi, 2-dimensi, dst) dan bentuk geometri dari masing-masing sel penyusunnya.

2. State Set

State set adalah himpunan keadaan atau status yang dapat dimiliki okeh masing-masing sel *cellular automata* tersebut.

3. Neighbourhood

Neighbourhood atau tetangga adalah sel-sel yang dapat mempengaruhi status sel pada *cellular automata*.Umumnya *neighbourhood* suatu sel hanya meliputi sel-sel yang berada disekitarnya (jari-jari *neighbourhood* – r, tidak besar).

2.5.2 Karakteristik *Cellular Automata*

Karakteristik *cellular automata* antara lain : sistem diskrit yang dinamis, *locality*, *parallelism* dan *emergent*. Dengan karakteristik ini, *cellular automata* sesuai digunakan untuk memodelkan sistem yang kompleks secara sederhana dan sesuai untuk diimplementasikan pada lingkungan parallel.

1. Sistem Diskrit yang Dinamis

Sistem diskrit yang dinamis adalah sistem yang memiliki spesifikasi berikut :

1. Memiliki entiti-entiti yang berubah seiring dengan berjalannya waktu. Perubahan ini disebabkan oleh sistem itu sendiri (karena faktor internal).
2. Entiti-entiti yang menyusun sistem tersebut terhitung (*countable*),
3. Perubahan entity-entiti itu terjadi dalam waktu yang diskrit (per *time-step*).

2. Locality

Locality berarti ketika sebuah sel berubah, status barunya hanya dipengaruhi oleh status lama dan status neighbours-nya. Karena umumnya *neighbourhood* suatu sel hanya meliputi sel-sel sekitarnya saja (jari-jari *neighbourhood* tidak besar) dapat dikatakan bahwa perubahan status dari tiap sel hanya bergantung pada dirinya dan sel-sel disekitarnya saja.

3. Parallelism

Parallelism berarti perubahan masing-masing sel dapat dilakukan dengan tidak bergantung pada sel lain sehingga semua sel dapat diperbaharui secara serentak. Karena itu *cellular automata* pada dasarnya sesuai diimplementasikan pada lingkungan parallel.

4. Emergent

Tiap sel penyusun *cellular automata* hanya melakukan fungsi-fungsi sederhana yang sepertinya tidak terlalu bermanfaat. Namun, ketika sel-sel tersebut dilihat sebai satu kesatuan, maka akan menjadi satu sistem yang dapat menghasilkan sesuatu yang besar. Jadi, seakan-akan sistem tersebut muncul dengan tiba-tiba (*emerges*), yaitu gabungan baian-bagiannya lebih besar daripada penjumlahan bagian-bagiannya. Hal inilah yang disebut *emergent behavior* dari *cellular automata.*

2.6 Pengujian Sistem

Pada tugas akhir ini dilakukan simulasi pengujian sistem steganografi dengan *noise* Gaussian, *noise* Salt & Pepper serta serangan geometris pada citra yang dikirim, seperti *rescaling* dan *cropping*. Hasilnya kemudian dianalisis apakah sistem sudah bekerja dengan baik atau belum dengan menggunakan parameter BER, PSNR, dan MOS.

2.6.1 *Noise* Gaussian

Noise Gaussian merupakan model *noise* yg mengikuti distribusi normal *standard* dengan rata-rata nol dan standard deviasi 1. Efek dari noise ini adalah munculnya titik-titik berwarna yg jumlahnya sama dengan persentase *noise.*

2.6.2 *Noise* Salt & Pepper

Noise Salt & Pepper merupakan model *noise* seperti taburan garam dan lada yang akan memberikan warna putih dan hitam pada titik yang terkena *noise*.

2.6.3 *Rescale*

Rescaling citra artinya adalah mengubah besarnya ukuran citra digital dalam *piksel* lalu dikembalikan ke ukuran semula. Adakalanya ukurannya diubah menjadi lebih kecil dari *file* aslinya dan adakalanya sebaliknya.

2.7 Performansi Sistem

Penilaian terhadap kualitas sistem ini dilakukan dengan penilaian obyektif dan subyektif.

2.7.1 Penilaian Obyektif[7]

Analisis kualitas citra digital berdasarkan penilaian obyektif dilakukan dengan menghitung nilai parameter PSNR *(Peak Signal to Noise Ratio)* dan BER *(Bit Error Rate)*.

2.7.1.1 *Peak Signal to Noise Ratio* (PSNR)

PSNR merupakan nilai perbandingan antara harga maksimum dari intensitas citra terhadap error citra yaitu MSE. Lebih jelasnya, MSE adalah nilai yang menyatakan rata-rata kuadrat *error*, dalam hal ini *error* menyatakan

selisih antar citra dimana kedua citra yang dibandingkan memiliki ukuran yang sama. Oleh karena itu, sebelum dapat menghitung PSNR suatu citra kita harus menghitung nilai MSE terlebih dahulu. Untuk menghitung nilai MSE digunakan persamaan berikut:

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad ...(2.5)$$

Sementara penghitungan nilai PSNR dilakukan dengan menggunakan persamaan berikut:

$$PSNR = 20 \log_{10} \frac{MAX_I}{\sqrt{MSE}} \quad ..... \quad (2.6)$$

PSNR yang semakin besar menandakan bahwa kualitas citra semakin bagus, hal ini karena *error* antara kedua citra semakin kecil.

2.7.1.2 *Character Error Rate* (CER)

CER merupakan persentase karakterpenyisipan yang mengalami *error* dengan jumlah keseluruhan karakter pada citra stego. Semakin kecil nilai CER, semakin bagus kualitas video, karena semakin kecil jumlah karakter pesan yang mengalami *error*.

CER dihitung dengan menggunakan rumus :

$$CER = \frac{I + S + D}{N} \times 100\% \quad .....(2.7)$$

Dimana I adalah jumlah karakter yang tersisipkan, S adalah jumlah pesan yang berubah, D adalah jumlah pesan yang terhapus, dan N adalah jumlah karakter maksimal.

2.7.1.3 *Avalanche Effect* [5]

Avalanche effect adalah karakteristik yang penting pada algoritma enkripsi. Parameter ini dapat kita lihat ketika kita mengganti satu bit pada *plain text* dan kemudian melihat perubahan bit setidaknya setengah dari bit dalam *cipher text*. Salah satu tujuan dari *avalanche effect* yaitu bahwa hanya dengan mengubah satu bit dapat terjadi perubahan besar, maka lebih sulit untuk para *attacker* melakukan analisis pada *chipper text*.

2.7.2 Penilaian Subyektif[3]

Penilaian subyektif adalah penilaian berdasarkan pengamatan mata manusia melihat seberapa bagus kualitas dari suatu citra, sehingga kualitas subyektif tergantung kepada persepsi visual pengamat.

Tabel 2.1 Kriteria MOS

Nilai	Kualitas Citra	Keterangan
1	Tidak Mirip	Tidak terlihat pula gambar asli pada citra yang diamati
2	Kurang Mirip	Citra yang diamati memiliki banyak noise, pada citra asli hampir tidak bisa terlihat
3	Cukup Mirip	Citra yang diamati memiliki noise, hanya terlihat sebagian citra asli
4	Mirip	Citra yang diamati memiliki sedikit noise, tapi citra asli masih bisa terlihat jelas dan mirip keseluruhan
5	Sangat Mirip	Citra yang diamati mempunyai kualitas yang sangat baik, hampir tidak ada bedanya dengan citra asli

Nilai rata-rata dari evaluasi MOS dihitung dengan rumus: $MOS = \frac{\sum_{i=1}^{n} R_i}{n}$ ..... (3.4)

di mana *n* adalah jumlah pengamat yang memberikan evaluasi terhadap kualitas citra steganografi. Dalam tugas akhir ini, n adalah sebanyak 30 orang.

3. Perancangan Dan Implementasi Sistem

3.1 Identifikasi Kebutuhan Sistem

Dalam perancangan sistem steganografi pada text serta menggunakan aturan 2 dimensi *Cellular Automata* untuk proses enkripsi, dibutuhkan beberapa spesifikasi dari perangkat keras *(hardware)* dan perangkat lunak *(software)* yang digunakan dalam penelitian tugas akhir ini.

3.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk mengimplementasikan sistem steganografi yang telah dirancang adalah sebagai berikut:

1. *System Model* : HP Probook 4420s
2. *Processor* : Intel® Core i3 350M 2.40 Ghz
3. *Memory* : 2048MB RAM

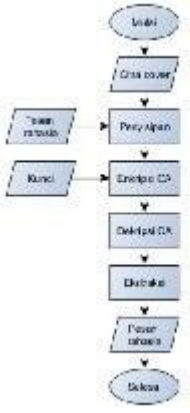
3.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk mengimplementasikan sistem steganalisis yang telah dirancang adalah sebagai berikut:

1. Sistem operasi Windows 7 Professional 32-bit
2. *Programming Tool* : Matlab R2009a
3. *Paint* untuk membuat ukuran gambar sesuai kebutuhan
4. Microsoft Office Excel 2007 untuk mengolah data hasil pengujian sistem
5. Microsoft Office Visio 2007 untuk membuat diagram blok dan diagram alir *(flow chart)*

3.2 Perancangan Sistem

Konfigurasi sistem yang dirancang pada tugas akhir ini terdiri dari 4 proses, yaitu proses penyisipan *text* untuk pengamanan level 1 dan proses enkripsi untuk pengamanan level 2 serta untuk proses pengambilan pesannya akan dilakukan dekripsi dan ekstraksi. Pada proses penyisipan menggunakan metode DCT dan pada tahap enkripsi menggunakan metode *2D Cellular Automata*.



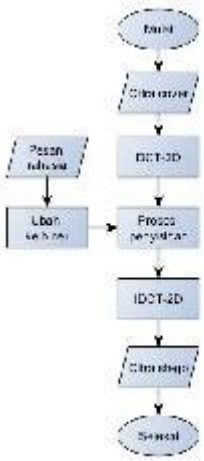
Gambar 3.1 Diagram Blok Sistem Secara Umum

Sistem ini secara umum dapat dijelaskan sebagai berikut:

1. Pesan rahasiaakan disisipkan pada citra cover melalui proses penyisipan, menghasilkan citra stego.
2. Citra stego akan dienkripsi menggunakan metode *Cellular Automata* (CA).
3. Di sisi penerima, citra stego akan didekripsi dan di ekstraksi yang hasilnya ialah pesan rahasi yang diinginkan.

3.2.1 Proses Penyisipan

Pada pengamanan level 1 akan menggunakan metode *Discrete Cosine Transform* (DCT). Pada proses ini pesan rahasia akan disisipkan pada suatucitra sehingga pesan rahasianya tidak akan kelihatan.



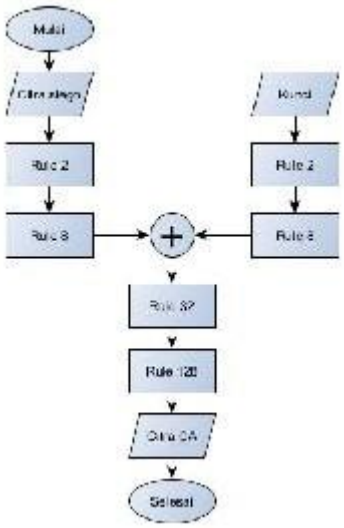
Gambar 3.2 Diagram Alir Proses Penyisipan Pesan Rahasia

Proses penyisipan ini dijelaskan sebagai berikut.

1. Setelah pengguna memilih sebuah citra RGB, kemudian diambil salah satu layer yang digunakan sebagai media penyisipan.
2. Pesan rahasia dituliskan dan diubah ke dalam biner untuk melakukan proses penyisipan ke dalam salah satu layer citra.
3. Kemudian dilakukan proses DCT blok 8x8 dan melakukan normalisasi terhadap layer yang telah diambil tadi dengan menggunakan kuantisasi level 50.
4. Kemudian mencari tempat penyisipan yang tepat pada layer dimana yang bukan bernilai 0 atau 1.
5. Setelah mendapatkan tempat penyisipan untuk bit-bit pesan rahasia, jika koefisien bernilai ganjil atau 1 dan bit pesan 0 maka nilai tempat penyisipan tersebut dikurangi satu (-1). Sedangkan jika koefisien bernilai genap atau 0 dan bit pesan 1 maka nilai pada tempat penyisipan tersebut ditambah 1 (+1).
6. Setelah proses penyisipan selesai layer tersebut di IDCT blok 8x8 dan dilakukan proses pembulatan terhadap nilai pikselnya.
7. Untuk tahap akhirnya layer tersebut digabungkan kembali dengan dua layer lainnya sehingga kembali kebentuk citra RGB dan disebut sebagai citra *cover*.
8. Proses penyisipan yang dilakukan bersifat *blind*, sehingga pada saat proses ekstraksi tidak dibutuhkan citra aslinya sebagai pembanding terhadap citra stegonya.

3.2.2 Proses Enkripsi

Pada pengamanan level 2 akan menggunakan metode *2D Cellular Automata. Rule* yang akan digunakan pada metode ini yaitu *rule* 2, *Rule* 8, *Rule* 32, dan *Rule* 128.



Gambar 3.3 Diagram Alir Proses Enkripsi *Cellular Automata*

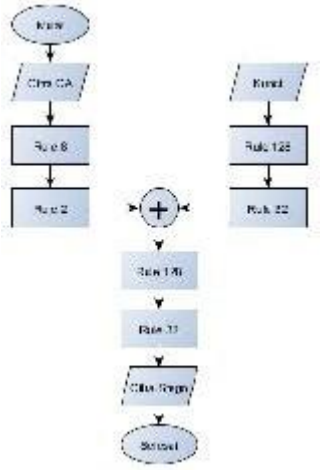
Proses penyisipan ini dijelaskan sebagai berikut :

1. Setelah proses penyisipan dilakukan, citra *cover* tersebut dipisah lagi menjadi 3 layer.
2. Kemudian diambil ukuran tiap dimensi layer untuk mendapatkan panjang karakter kunci ini dimaksudkan untuk mencocokkan ukuran citra stego dan citra kunci.
3. Teks kunci yang dituliskan diubah kebentuk matriks dan dibentuk menjadi citra sehingga dapat digabungkan dengan citra stego. Citra kunci tersebut dibentuk menjadi 3 layer.
4. Setelah mendapatkan tiap layer citra kunci tadi kemudian dilakukan proses *rule* 2 dan *rule* 8, juga pada tiap layer citra stego dilakukan proses *rule* 2 dan *rule* 8, pada *rule* 2 geser ke kanan sedangkan *rule* 8 geser ke bawah.
5. Proses selanjutnya, hasil dari kedua rule tersebut tiap layer yang sama kemudian di xor dan hasilnya dilakukan lagi proses *rule* 32 dan *rule* 128. Pada *rule* 32 geser ke kiri sedangkan *rule* 128 geser ke atas.

6. Tiap layer yang sama tersebut kemudian digabungkan yang akan menghasilkan gambar yang rusak sehingga tidak dapat diketahui gambar aslinya.
7. Pada *rule-rule* tersebut digeser beberapa kali agar menghasilkan gambar yang benar-benar tidak dapat diketahui lagi.

3.2.3 Proses Dekripsi

Proses dekripsi merupakan kebalikan dari proses enkripsi, yang dimaksudkan untuk mendapatkan kembali citra stego.



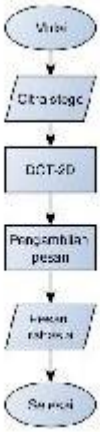
Gambar 3.4 Diagram Alir Proses Dekripsi *Cellular Automata*

Proses dekripsi dijelaskan sebagai berikut :

1. Penerima menerima citra CA dari pengirim dalam bentuk gambar yang telah teracak, kemudian sisi penerima menentukan CA *rule* yang digunakan untuk melakukan dekripsi citra CA tersebut.
2. Penerima juga harus mengetahui kunci yang digunakan untuk menjalankan proses dekripsi tersebut sehigga proses CA dapat dilakukan
3. Citra CA yang terekripsi pertama diberikan *Cellular automata rule* 8 karena proses CA yang terakhir yang digunakan adalah *Cellular automata rule* 128 yang merupakan hasil transpose CA *rule* 8.
4. Setalah itu citra CA diberikan lagi *Cellular automata rule* 2 yang merupakan *transpose rule* 32
5. Kunci yang diberikan pada citra juga diberikan CA *rule* dimana pertama kali diberikan CA *rule* 128 kemudian CA *rule* 32
6. Setelah citra dan kunci selesai di CA hasil dari CA kemudian di xor sehigga kunci dan citra dapat digabung.
7. Setelah itu hasilnya diberikan kembali CA *rule* 128 dan CA *rule* 32
8. Kemudian didapatkan citra stego yang merupakan sebuah citra yang berisi pesan rahasia.

3.2.4 Proses Ekstraksi

Proses ekstraksi pesan merupakan proses kebalikan dari proses penyisipan pesan, pada proses ini pesan akan diambil kembali dari citra stego.



Gambar 3.5 Diagram Alir Proses Ekstraksi Pesan Rahasia

Proses dekripsi dijelaskan sebagai berikut :

1. Pertama kita harus mengetahui jumlah karakter yang dikirimkan.
2. Setelah itu kita mengambil salah satu layer dari citra stego di mana tempat pesan itu disisipkan, setelah itu layer tersebut di DCT.
3. Kemudian dilakukan pengambilan bit dari layer tersebut yang merupakan pesan rahasia yang dikirimkan.
4. Bit tersebut diproses untuk diubah kembali menjadi karakter yang isinya adalah pesan rahasia.

4. Pengujian Dan Analisis Sistem

Pada bab ini dibahas mengenai pengujian dan performansi sistem. Pada pengujian sistem dilakukan simulasi proses steganografi dengan menberikan *noise* Gaussian, *noise* Salt & Pepper serta serangan geometris pada citra yang dikirim, seperti *rescaling*. Hasilnya kemudian dianalisis apakah sistem sudah bekerja dengan baik atau belum dengan menggunakan parameter CER, PSNR, dan MOS.

4.1 Lingkup Pengujian

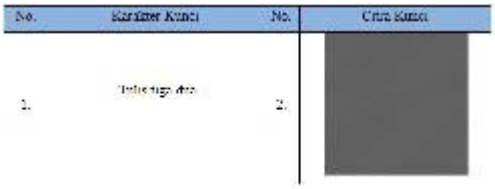
Pengujian pada tugas akhir ini menggunakan 1 buah citra RGB dengan ukuran piksel yang berbeda-beda dengan format Bitmap, yaitu citra yang berukuran 64x64, 128x128, 256x256, 512x512 piksel, 14 karakter huruf yang digunakan sebagai kunci yang dibentuk menjadi citra rahasia, serta berbagai karakter pesan yang menjadi pesan rahasia. Berikut ini merupakan gambar dari citra *cover* dan citra kunci yang digunakan dalam pengujian:

4.1.1 Citra Cover

Tabel 4.1 Citra Pengujian

4.1.2 Pesan Kunci

Tabel 4.2 Citra Kunci



4.2 Skenario Pengujian Sistem

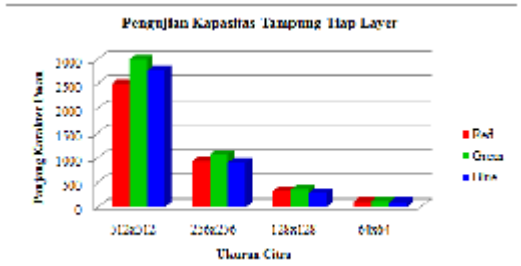
Skenario yang digunakan dalam pengujian adalah sebagai berikut:

1. Melakukan proses penyisipan pesan rahasia pada citra *cover* dengan metode DCT-2D dan Cellular *Automata*.
2. Menghitung jumlah maksimum karakter yang dapat di tempati oleh masing-masing layer *red*, *green* dan *blue*.
3. Menguji pengaruh besarnya banyaknya pesan rahasia dan besarnya citra *cover*.
4. Memberikan berbagai jenis gangguan (*noise* Gaussian, *noise* Salt & Pepper, dan *rescaling*) pada citra stego.
5. Menghitung waktu komputasi dari tiap ukuran citra yang diproses dan juga akurasi dari pesan yang dikirim.
6. Mengukur kualitas citra dengan menggunakan CER dan PSNR pada sistem.
7. Membandingkan citra *cover* asli dengan citra stego dan citra CA untuk mendapatkan nilai secara subyektif atau *Mean Opinion Score* (MOS) menggunakan sistem survei kepada 30 orang pengamat.

4.3 Analisis Data Hasil Pengujian Sistem

Berdasarkan skenario pengujian yang telah ditetapkan sebelumnya, maka dilakukan analisis sebagai berikut:

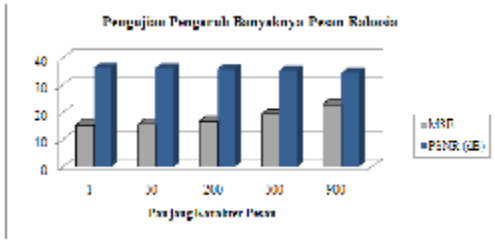
4.3.1 Pengujian Kapasitas Tampung Pada Tiap Layer



Gambar 4.1 Grafik Panjang Karakter Maksimal Tiap Layer

Berdasarkan tabel dan grafik di atas, dapat kita lihat kapasitas tampung untuk layer *green* lebih besar dari kedua layer lainnya. Sedangkan untuk yang paling kecil adalah layer *blue* yang nilainya sebagian besar lebih kecil dibandingkan layer lainnya.

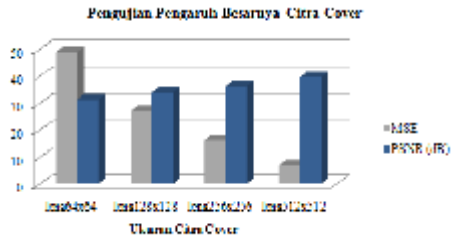
4.3.2 Pengujian pengaruh Banyaknya Pesan Rahsia



Gambar 4.2 Grafik Nilai MSE dan PSNR pada Citra Stego dengan panjang pesan berbeda

Dapat dilihat dari tabel dan grafik di atas bahwa nilai MSE adalah 15,36 dan nilai PSNR adalah 36,26 dB pada citra *cover* lena256x256 yang disisipkan panjang pesan 1 karakter dan pada citra yang sama disisipkan panjang pesan 900 karakter didapatkan nilai MSE 22,9 dan nilai PSNR 34,53 dB. Hal ini menunjukkan bahwa penyisipan yang dilakukan dengan panjang pesan yang berbeda-beda menghasilkan MSE dan PNSR yang berbeda pula. Semakin sedikit jumlah karakter pesannya maka nilai MSE akan semakin kecil dan nilai PNSR akan semakin besar, begitu pula sebaliknya.

4.3.3 Pengujian Pengaruh Besarnya Citra *Cover*



Gambar 4.3 Grafik Nilai MSE dan PSNR pada Citra Stego dangan Citra *Cover* berbeda

Dapat dilihat dari tabel dan grafik di atas bahwa nilai MSE adalah 48,94 dan nilai PSNR adalah 31,23 dB pada citra *cover* lena64x64 yang disisipkan panjang pesan 100 karakter dan pada panjang pesan yang sama disisipkan pada citra *cover* lena512x512 didapatkan nilai MSE 7,01 dan nilai PSNR 39,66 dB. Hal ini menunjukkan bahwa penyisipan yang dilakukan dengan pada citra *cover* yang berbeda-beda menghasilkan MSE dan PNSR yang berbeda juga. Semakin kecil ukuran citra *cover* maka nilai MSE akan semakin besar dan nilai PNSR akan semakin kecil, begitu pula sebaliknya.

4.3.4 Pengujian Akurasi Tiap Layer

Tabel 4.6 Nilai CER

Layer	Ukuran Citra Cover	Jumlah Karakter yang Disisipkan	CER (%)
Red	512x512	2481	0,08
	256x256	896	0,08
	128x128	282	13,04
	64x64	89	12,35
Green	512x512	2481	1,6
	256x256	896	27,45
	128x128	282	29,78
	64x64	89	42,69
Blue	512x512	2484	0
	256x256	896	0
	128x128	282	0
	64x64	89	0

Pada tabel di atas menunjukkan nilai CER pada layer *blue* yaitu 0%, ini menunjukkan bahwa pada layer ini sangat baik sebagai tempat penyisipan pesan karena tidak ada kesalahan sama sekali ketika pesan diekstrak. Sedangkan untuk layer *red* dan *green* masih ada kesalahan yaitu sekitar 7,98% dan 25,35%.

4.3.5 Pengaruh Pemberian *Noise* Gaussian, *Noise* Salt & Pepper dan *Rescale*

Tabel 4.7 Pengaruh Pemberian *Noise* Gaussian

Layer Citra	Pesan		CER (%)
	Ukuran Pesan	Jumlah Karakter yang Disisipkan	
Red	512x512	2484	99,37
	256x256	896	99,3
	128x128	282	100
	64x64	89	100
	512x512	2484	99,15
Green	256x256	896	99,8
	128x128	282	99,37
	64x64	89	98,31
	512x512	2484	99,49
Blue	256x256	896	99,8
	128x128	282	100
	64x64	89	100

Tabel 4.8 Pengaruh Pemberian *Noise* Salt & Pepper

Layer Citra	Pesan		CER (%)
	Ukuran Pesan	Jumlah Karakter yang Disisipkan	
Red	512x512	2484	99,64
	256x256	896	99,2
	128x128	282	99,38
	64x64	89	100
	512x512	2484	99,28
Green	256x256	896	99,4
	128x128	282	98,73
	64x64	89	100
	512x512	2481	99,57
Blue	256x256	896	99,8
	128x128	282	100
	64x64	89	98,31

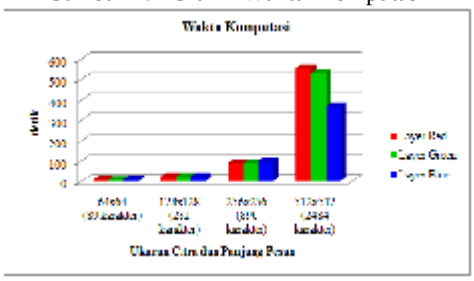
Tabel 4.9 Pengaruh Penggunaan *Rescale*

Layer Citra	Layer Citra			
	Ukuran Pesan	Jumlah Karakter yang Disisipkan	Ukuran Pesan Setelah Rescale	CER (%)
Red	512x512	2484	384x384	49,71
	256x256	896	192x192	97,82
	128x128	282	96x96	99,1
	64x64	89	48x48	99,87
Green	512x512	2484	384x384	65,39
	256x256	896	192x192	95,32
	128x128	282	96x96	98,58
	64x64	89	48x48	98,9
Blue	512x512	2481	384x384	66,1
	256x256	896	192x192	88,14
	128x128	282	96x96	92,25
	64x64	89	48x48	95,5

Berdasarkan tabel hasil pengujian *noise* Gaussian dan *noise* Salt & Pepper di atas, dapat disimpulkan bahwa pesan yang disisipkan tidak dapat terekstrak dengan baik. Pemberian *noise* menyebabkan perbedaan bit citra stego yang cukup signifikan yang menyebabkan sistem mendeteksi bahwa tidak ada pesan rahasia yang disisipkan pada citra sehingga pesan yang disisipkan tidak dapat terekstrak. Dengan demikian, dapat dikatakan bahwa sistem yang dibuat sangat rentan kehilangan pesan rahasia jika terkena *noise* dan *rescale*.

4.3.6 Pengaruh Ukuran Citra Cover terhadap Waktu Komputasi

Gambar 4.4 Grafik Waktu Komputasi



Sesuai tabel hasil komputasi di atas, pada sistem ini proses komputasi membutuhkan waktu yang cukup lama untuk ukuran citra 512x512 untuk semua layer. Sedangkan untuk citra yang berukuran kecil tidak membutuhkan waktu lama untuk melakukan komputasi.

4.3.7 Pengujian Metode Enkripsi menggunakan *Avalenche Effect*

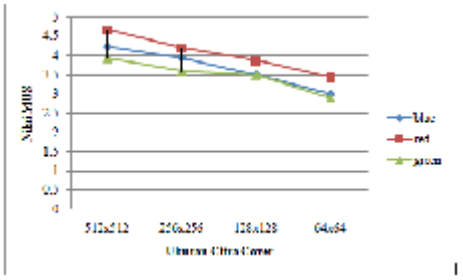
Tabel 4.11 Pengujian *Avalanche Effect* dengan mengganti satu bit dari kunci

Ukuran Citra	Pesan	Kunci	Total Bit Beda	Avalanche Effect (%)
Lena512x512	Telkom University	tulis tgs dsc	26179	0,8728
Lena256x256	Telkom University	tulis tgs dsc	14048	0,8918
Lena128x128	Telkom University	tulis tgs dsc	3510	0,8926
Lena64x64	Telkom University	tulis tgs dsc	877	0,8911

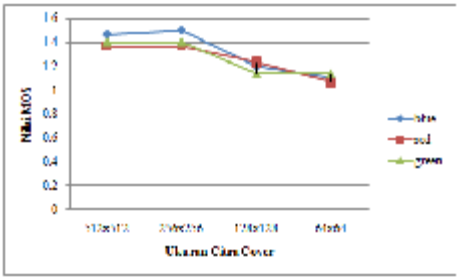
Berdasarkan Tabel 4.11 di atas bahwa dengan perbedaan satu bit saja pada dua buah data dapat menghasilkan jumlah bit beda tergantung dari ukuran citranya. Semakin besar ukuran citranya berarti semakin banyak bit yang ada maka total bit yang berbeda juga semakin besar, sedangkan semakin kecil kecil ukuran citranya berarti semakin kecil bit yang ada sehingga total bit yang berbeda

semakin kecil. Namun, nilai rata-rata *Avalanche Effect* hanya 0,8925 %.

4.3.8 Pengujian Berdasarkan MOS (Mean Opinion Score)



Gambar 4.5 Grafik Nilai MOS untuk Citra Stego 1



Gambar 4.7 Grafik Nilai MOS untuk Citra CA 1

Berdasarkan Gambar 4.5 di atas, telihat bahwa hasil yang diperoleh dari 30 responden menunjukkan kualitas dari layer *red* dan *blue* pada stego image lebih baik dibanding layer *green*. Terbukti untuk citra stego dengan maksimum karakter nilai MOS layer *red* dan *blue* di atas 4 sedangkan citra stego layer *green* hanya mendekati angka 4.

Berdasarkan Gambar 4.7, ketika nilai MOSnya sangat rendah itu berarti kualitas dari sistem yang dibuat sangat baik, karena pada dasarnya sistem CA dibuat bertujuan agar citra stegonya tidak dapat dikenali lagi.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pengujian yang dilakukan pada Tugas Akhir ini, dapat disimpulkan sebagai berikut.

1. Sistem yang dibuat mampu menyembunyikan pesan rahasia pada citra *cover* disemua layer.
2. Panjang pesan dan besarnya ukuran citra *cover* sangat berpengaruh terhadap nilai MSE, PSNR dan juga waktu komputasinya. Semakin besar citra *cover* dan pesan rahasianya maka semakin lama waktu komputasinya.
3. Akurasi dari sistem yang dibuat untuk penyisipan pada layer *blue* yaitu 100% sedangkan pada layer *red* dan *green* yaitu 92,02% dan 74,65%.
4. Sistem tidak cukup baik dalam menanggapi serangan berupa *noise* Gaussian, *noise* Salt & Pepper, terbukti dengan nilai CER yang hampir mencapi 100%. Untuk penggunaan *rescale* sebesar 75%, hasilnya juga kurang baik yaitu di atas 80%.
5. Berdasarkan pengujian *Avalanche Effect* algoritma enkripsi yang dibuat masih rentan terkena serangan, karena jumlah bit yang berbeda tidak sampai 50% bahkan hanya mendekati 1%.
6. Berdasarkan nilai MOS perbandingan citra asli dan citra stego cukup baik, juga untuk perbandingan citra asli dan citra CA mendapatkan nilai yang rendah, itu berarti proses enkripsi yang dilakukan untuk membuat gambarnya rusak berhasil.

5.2 Saran

Adapun saran untuk pengembangan tugas akhir selanjutnya adalah

1. Melakukan penyempurnaan agar saat diberikan *attack*, pesan rahasia yang ada di dalamnya masih dapat di ekstrak.
2. Menggabungkan metode *Cellular Automata* dengan metode steganografi lainnya.
3. Menggunakan media video sebagai *cover*.
4. Menggunakan format citra yang lain seperti .jpg, .gif, atau .png.

DAFTAR PUSTAKA

[1] Anandika, Hari. 2012. *Perancangan dan Analisis Multiple Watermarking pada Citra Digital berbasis Iterative Threshold dan Deteksi Tepi*. Bandung: Institut Teknologi Telkom.

[2] Buchholz, Jorg J. 2001. *Matlab Implementation of the Advanced Encryption Standard*.

[3] C. Gupta, Prakash. 2006. *Data Communication and Computer Network.* New Delhi : Asoke K. Ghosh

[4] Komputer, Wahana. 2003. *Memahami Model Enkripsidan Security Data*. Yogyakarta: Andi.

[5] Kumar, Amish. 2013. *Effective Implementation and Avalanche Effect of AES*. Bhopal : International Journal of Security, Privacy and Trust Management (IJSPTM).

[6] Mulyantini, Agustien. 2012. *Analisis Steganografi pada Citra Digital menggunakan DCT (Discrete Cosine Transform) dan Enkripsi AES*. Bandung: Institut Teknologi Telkom.

[7] Munir, Rinaldi. 2006. *Kriptografi*. Bandung: Penerbit Informatika

[8] Putra, Darma. 2010. *Pengolahan* Citra Digital. Yogyakarta: Andi.