Implementasi *Reversible Data hiding* dengan Menggunakan *Difference Histogram Shiffting* pada *Medical Image* Terenkripsi

Proposal Tugas Akhir

Kelas TA 1

Muhammad Fadhlan Putranto
NIM: 1301140418

Program Studi Sarjana Teknik Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2017

Lembar Persetujuan

Implementasi *Reversible Data hiding* dengan Menggunakan *Difference Histogram Shiffting* pada *Medical Image* Terenkripsi

The Implementation of Reversible Data Hiding Using Difference Histogram Shifting in Encrypted Medical Image

Muhammad Fadhlan Putranto
NIM: 1301140418

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada
Program Studi Sarjana Teknik Informatika
Fakultas Informatika Universitas Telkom

Bandung, 22 April 2017
Menyetujui

Calon Pembimbing 1	Calon Pembimbing 2
Ari M Barmawi, M.Sc.,Ph.D.	Bambang Ari Wahyudi, S.Kom., M.T.
NIP: 08600459-4	NIP: 14860086-1

Abstrak

Penggunaan metode *reversible data hiding* (RDH) pada gambar terenkripsi dengan menggunakan metode klasik belum bisa dilakukan secara langsung. Selain itu penggunaan RDH pada media gambar yang membutuhkan ketelitian yang sangat tinggi seperti gambar medis sangat berbahaya dan dapat mempengaruhi interpretasi dari konten gambar itu sendiri.

Pada penelitian ini diusulkan sistem yang dapat melakukan enkripsi gambar dan penyisipan pesan pada media gambar yang terenkripsi tanpa mempengaruhi atau merusak kontent gambar setelah pesan didekripsi sehingga gambar dapat dipulihkan kembali seperti semula. Penelitian ini mengusulkan kerangka baru dalam melakukan enkripsi gambar agar pesan dapat disisipkan secara langsung pada gambar yang telah dienkripsi. Dengan menggunakan kerangka yang diusulkan dalam mengenkripsi gambar, maka semua skema RDH sebelumnya dapat diaplikasikan dalam melakukan penyisipan pesan karena pada penelitian ini skema RDH yang diusulkan bersifat independen dari enkripsi gambar.

Kata Kunci: *Riversible Data Hiding, Difference Histogram Shifting, Gambar Terenkripsi.*

Daftar Isi

Abstrak **i**

Daftar Isi **ii**

I Pendahuluan **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Tujuan 2
1.4 Hipotesis 2
1.5 Rencana Kegiatan 2
1.6 Jadwal Kegiatan 3

II Kajian Pustaka **5**
2.1 *Reversible Data Hiding* 5
2.2 *Difference Histogram Shifting* 6
2.3 Algoritma *Stream Chipper* RC4 8
2.3.1 Inisialisasi *State Array* 8
2.3.2 Algoritma *Key-scheduling* 8

III Algoritma **9**
3.0.1 Pembangkitan Kunci Enkripsi 9
3.1 Algortima Permutasi Gambar 10
3.2 Pengukuran Kualitas Citra 11
3.2.1 *Mean Square Error*(MSE) 11
3.2.2 *Peak Signal Noise Ratio* (PSNR) 11

IV Metodologi dan Desain Sistem **12**
4.1 Gambaran Umum Sistem 12
4.2 Penjelasan Rancangan Sistem 13
4.2.1 Enkripsi Gambar 14
4.2.2 Penyisipan Data 16
4.2.3 Dekripsi Gambar dan Ekstraksi Data 17
4.3 Gambaran Singkat Sistem 17

Daftar Pustaka **18**

Lampiran **19**

V Algoritma **20**

Bab I

Pendahuluan

1.1 Latar Belakang

Data Hiding atau penyembunyian data merupakan skema dalam media digital yang digunakan untuk menyisipkan data sekunder menjadi informasi asli. Penyembunyian data dalam kriptografi dan steganografi sangat berbeda. Pada kriptografi, data yang disandikan (*chipertext*) tetap ada dan keberadaanya dapat diketahui, sedangkan pada steganografi, *chipertext* dapat disembunyikan sehingga pihak ketiga tidak mengatahui keberadaanya. Oleh karena itu, pada era digital seperti sekarang ini, keamanan suatu informasi akan memiliki nilai lebih tinggi karena akan menyangkut aspek-aspek keputusan bisnis ataupun kepentingan umum. Sehingga, steganografi semakin dibutuhkan guna memberikan tingkat keamanan yang maksimal dalam proses pengiriman informasi.

Steganografi merupakan salah satu cara penyembunyian data rahasia di dalam media lain, seperti gambar, audio maupun video dimana orang lain tidak mengetahui bahwa terdapat data rahasia yang tersembunyi di dalam media tersebut, kecuali bagi pihak-pihak yang memiliki kepentingan atau hak akses. Penelitian kali ini lebih mengfokuskan kepada penerapan *reversible data hiding* pada media terenkripsi.*Revesible data hiding* (RDH) merupakan teknik penyisipan data ke dalam media gambar yang nantinya data yang disembunyikan dapat diambil atau diekstrak dari dalam gambar dan yang terpenting dari teknik ini adalah media gambar (*image cover*) dapat direkonstruksi atau dipulihkan kembali seperti gambar aslinya.

Reversible data hiding memiliki manfaat ketika ketelitian sangat diperhatikan,seperti gambar medis *(medical image)*. Akan tetapi, dengan melakukan perubahan secara sengaja pada *medical image* sangat berbahaya dan dapat mempengaruhi interpretasi dari konten. Misalnya, perubahan yang tidak disengaja dari gambar X-ray dapat mengakibatkan misdiagnosa sehingga dapat mengakibatkan tindak pidana dan hukum yang dapat merugikan berbagai pihak. Oleh karena itu RDH dirancang agar dapat memecahkan masalah tersebut.

Pada penelitian sebelumnya [3],banyak skema *reversible datang hiding* yang

telah diajukan, seperti *lossless comparison*[1], *difference expansion* (DE)[6], dan *histogram shiffting* (HS)[5]. Akan tetapi, penyisipian pesan pada media gambar yang terenkripsi secara langsung tidak dapat dilakukan karena setelah dilakukan penyisipan atau enkripsi, korelasi antar pixel tetangga akan hilang sehingga algoritma RDH perlu dirancang secara khusus agar dapat menyisipkan pesan pada domain terenkripsi secara langsung. Untuk mencapai RDH pada media terenkripsi secara langsung, maka disini diajukan skema RDH baru yang lebih efektif dalam melakukan RDH pada domain tereknripsi.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang ingin saya angkat adalah bagaimana caranya agar korelasi antar pixel tetangga/ koefesien tetap ada setelah domain terenkripsi sehingga pesan atau data dapat disisipkan pada domain terenkripsi secara langsung.

1.3 Tujuan

pada tugas akhir ini, penulis akan mengimplementaskan kerangka *revesible data hiding* yang diusulkan dalam mengenkripsi gambar dan menerapkan skema RDH dengan pendekatan *difference histogram shifing* (DHS) dalam melakukan penyisipan data pada domain terenkripsi.

1.4 Hipotesis

berdasarkan studi literatur [3] yang telah dilakukan, dengan menerapkan spesifik algoritma enkripsi dalam mengenkripsi gambar maka korelasi antar pixel tetangga dapat bertahan. Dengan demikian, pendekatan DHS pada skema RDH dapat diterapkan dalam melakukan penyisipan pesan yang bersifat reversible pada *host image* dengan memodifikasi perbedaan histogram.

1.5 Rencana Kegiatan

Rencana kegiatan yang akan saya lakukan adalah sebagia berikut:

- Studi Literatur

  Studi literatur adalah proses mengkaji bidang yang akan dilakukan pada penelitian. Tahap pertama yang dilakukan adalah memahami prinsip-prinsip dasar dari *riversible data hiding* dan *difference histogram shifting*. Dalam hal ini dilakukan pengumpulan dan pembelajaran lebih mendalam dari literatur yang berkaitan dengan RDH dan DHS. Literatur yang digunakan berasal buku,paper, jurnal serta Tugas Akhir yang terkait dan telah diselesaikan.

- Pengumpulan Data

  Data yang digunakan pada penelitian ini berupa citra X-ray atau citra medis lainnya yang didapat dari internet atau dari arsip rumah sakit.

Citra tersebut akan diambil dari berbagai ukuran dan kualitas yang berbeda yang nantinya akan digunakan sebagai sampel penelitian.

- Analisis Kebutuhan Sistem

  Proses analisis kebutuhan sistem baik kebutuhan perangkat keras maupun perangkat lunak untuk sistem untuk melakukan enkripsi maupun dekripsi gambar dan penyisipan pesan.

- Perancangan Sistem

  Sistem akan dirancang untuk melakukan enkripsi gambar dengan menggunakan algoritma enkripsi . Lalu gambar yang telah dienkripsi akan dilakukan penyisipan pesan dengan menggunakan metode DHS.

- Implementasi Sistem

  Sistem yang sudah dirancang, kemudian diimplementasikan yang terdiri dari empat proses yaitu enkripsi gambar, dekripsi gambar, penyisipan pesan dan ekstraksi pesan.Implementasi ini dilakukan sesuai pada rancangan skema yang telah dibuat sebelumnya.

- Pengujian Sistem dan Analisis

  Pengujian dilakukan untuk mengetahui sistem yang dibangun sudah sesuai dengan rancangan atau belum. Selain itu, pengujian juga dilakukan untuk menganalisa efektitas dari sistem dalam menangani masalah yang ada dengan menggunakan komponen terukur yaitu nilai PSNR dan kapasitas penyisipan.Hasil yang telah didapat dari penelitian kemudian direpresentasikan dalam bentuk tabel dan grak.

- Pemuatan Laporan Tugas Akhir

  Laporan dibuat untuk mendokumtasikan segala kegiatan dan hasil yang didapat dari pengerjaan Tugas Akhir ini.

1.6 Jadwal Kegiatan

Laporan proposal ini akan dijadwalkan sesuai dengan tabel yang diberikna berikutnya.

Tabel 1.1: Jadwal kegiatan proposal tugas akhir

No	Kegiatan	Bulan ke-																					
		1				2				3				4				5				6			
1	Studi Literatur																						
2	Pengumpulan Data																						
3	Analisis dan Perancangan Sistem																						
4	Implementasi Sistem																						
5	Analisa Hasil Implementasi																						
6	Penulisan Laporan																						

Bab II

Kajian Pustaka

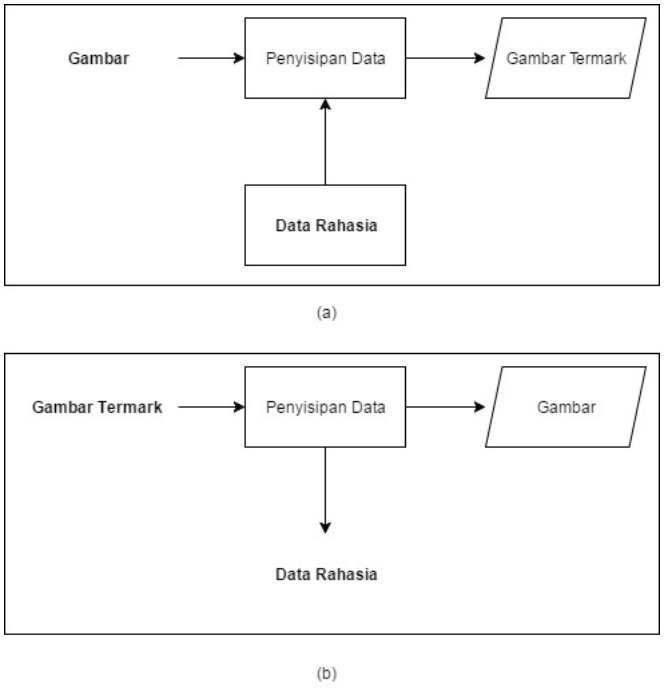
2.1 *Reversible Data Hiding*

Reversible data hiding (RDH) dapat menyembunyikan data ke dalam gambar digital dan yang terpenting gambar aslinya dapat rekonstruksi kembali setelah pesan yang disisipkan diekstrak[3].

RDH secara umum terdiri dari dua bagian. Bagian Pertama adalah penyisipan data seperti yang terlihat pada gambar 2.1(a) dan bagian lainnya adalah ekstraksi data dan pemulihan gambar asli seperti yang terlihat pada gambar 2.1(b). Pada prosesnya, penyisipan data dilakukan dengan berbagai teknik yang bersifat *reversible* atau yang dapat dibalikan. Penelitian tentang teknik RDH sudah banyak dilakukan sebelumnya seperti *Least Significant Bit* (LSB) [4], *difference expansion*(DE)[6], *histogram shifting*(HS)[5] dan lain-lain.*Reversible data hiding* dapat diaplikasikan pada gambar yang membutuhkan ketelelitian yang sangat tinggi seperti gambar medis, militer dan gambar satelit. *Reversible data hiding* digunakan agar media gambar yang telah disisipkan pesan dapat dipulihkan kembali menjadi gambar seperti aslinya sebelum gambar tersebut disisipkan. Untuk melakukan penyisipan data dibutuhkan masukan,antara lain;

- Media Gambar
- Data atau pesan
- Kunci

Keluaran dari skema ini adalah media gambar yang tersisipkan pesan dan bersifat *reversible*.



Gambar 2.1: Blok Diagram *reversible data hiding* (a)Penyisipan Data (b)Ekstraksi Data

2.2 *Difference Histogram Shifting*

Difference histogram Shifting (DHS) merupakan salah satu pendekatan skema RDH berbasis *histogram shifting*. DHS memiliki kelebihan dibandingkan skema RDH lainnya, yaitu kapasitas penyisipan data yang lebih besar dan dapat mempertahankan kualitas visualnya.

Ide utama dari metode DHS ini adalah mengeksplore kolerasi antar pixel tetangga pada citra.Mengikuti blok diagram 2.2, langkah pertama adalah membuat *difference image D(i,j)* berukuran $M\ X\ \frac{N}{2}$ dari gambar original *I(i,j)* berukuran $M\ X\ N$ mengikuti persamaan 2.1 dan algoritma 1.

$$D(i,j) = I(i,2j+1) - I(i,2j) \tag{2.1}$$

dimana *I(i,2j+1)* dan *I(i,2j)* merupakan kolom ganjil dan kolom genap. Langkah kedua adalah melakukan pergeseran *histogram* pada *difference image*. Jika nilai selisihnya bernilai lebih dari 2 maka kolom ganjil ditambah satu dan jika lebih kecil dari -2 maka kolom ganjil dikurang dengan 1. Lalu *difference image* yang telah dimodifikasi direperesentasikan dengan

$$\tilde{D}(i,j) = \tilde{I}(i,2j+1) - I(i,2j) \tag{2.2}$$

dimana

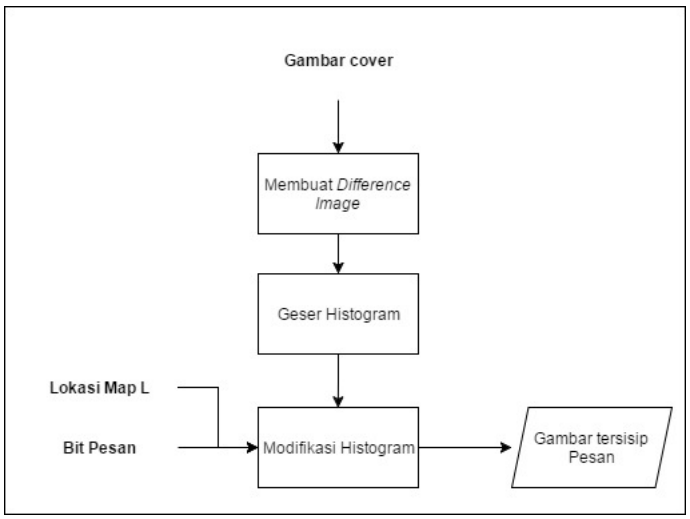
$$\tilde{I}(i,2j+1) = \begin{cases} I(i,2j+1) + 1, & jikaD(i,j) \geq 2 \\ I(i,2j+1) - 1, & JikaD(i,j) \leq -2 \\ I(i,2j+1), & lainnya \end{cases} \tag{2.3}$$

Dengan demikian, maka dapat dihasilkan perbedaan histogram (*difference histogram*) dengan nilai ± 1 dan 0. Sehingga penyisipan pesan dapat dilakukan dengan mengikuti persamaan 4.4.

Algorithm 1 Prosedur inisialisasi *difference image*.

1: **procedure** CREATEDIFFERENCEIMAGE(I)
2: Start
3: **For** $i = 0 : M - 1$ **do**
4: **For** $j = 0 : \frac{N}{2} - 1$ **do**
5: $D[i][j] = I[i][2j + 1] - I[i][2j]$
6: **EndFor**
7: **EndFor**
8: **return** ***D***
9: **end procedure**

Tujuan dari DHS ini adalah agar pesan atau data dapat disisipkan ke dalam citra atau gambar dengan mengmodifikasi perbedaan histogram. Selain itu,agar dapat mengidentifikasi daerah atu pixel yang mengalami perubahan. Masukan pada metode ini adalah media gambar(*original/watermarked image*),



Gambar 2.2: Blok Diagram *DHS*

Kunci dan bit pesan dan lokasi map L.

Keluaran dari metode ini adalah gambar yang tersisip pesan (pada proses penyisipan) dan bersifat *reversible*.

2.3 Algoritma *Stream Chipper* RC4

Algoritma RC4(*Rivest chiper* 4) merupakan algoritma kriptografi simetrik karena menggunakan kunci yang sama untuk mengenkripsi maupun dekripsi suaru pesan, data ataupun informasi. RC4 adalah sebuah *sychrone stream chiper*, yaitu mengenkripsi plainteks secara digit per digit atau byte per byte pesan dengan cara mengkobinasikan dengan operasi biner(XOR) dengan angka acak. Algoritma RC4 digunakan untuk menghasilkan kunci yang bersifat acak yang nantinya diopersikan dengan operasi XOR terhadap bit-bit. Pada prosesnya, algoritma RC4 terbagi menjadi, pertama inisialisasi *state-array*, *key-scheduling algorithm* (KSA) dan penghasilan kunci enkripsi menggunakan *pseudo-random generation algorithm* (PRGA).

2.3.1 Inisialisasi *State Array*

Dalam penginisialisasian, terdapat 2 *state-array* yang harus diinisialisasi, S dan K. Array S dan K memiliki panjang 256. Array S dinisialisasi angka 0 sampai dengan 255. Sedangkan array K diinisialisasi dengan kunci secara berulang sampay seluruh array terisi penuh. Algoritmanya seperti contoh pada algoritma 8 berikut ini :

Algorithm 2 Prosedur inisialisasi *state-array*.

```
1: procedure INISIALISASISTATEARRAY
2:     Start
3:     For i = 0 : 255 do                          ▷ Pemberian nilai awal
4:         S[i] = i
5:         K[i] = key[imodkey.length()]
6:     EndFor
7: end procedure
```

2.3.2 Algoritma *Key-scheduling*

Algoritma *key-scheduling* digunkan untuk menghasilkan permutasi dari array S yang sudah terisi tadi berdasarkan kunci(array K) yang tersedia. Array S[i] dengan S[j] akan ditukar berdasarkan nilai i dan j. Algoritmanya seperti contoh algoritma 9.

Bab III

Algoritma

Algorithm 3 Prosedur inisialisasi *state-array*.

```
1: procedure INISIALISASISTATEARRAY
2:     Start
3:     For i = 0 : 255 do                          ▷ Pemberian nilai awal
4:         S[i] = i
5:         K[i] = key[i mod key.length()]
6:     EndFor
7: end procedure
```

3.0.1 Pembangkitan Kunci Enkripsi

Setelah memiliki *state-array* yang teracak, maka selanjutnya adalah melakukan PRGA untuk menghasilkan kunci enkripsi yang nantinya digunakan untuk di-XOR-kan dengan *plain*. Pertama *incremen* nilai i dan dapatkan nilai j dari hasil j ditambahkan dengan array S ke-i, lalu tukar S[i] dan S[j]. lalu simpan *byte* kunci ke dalam B[n] ,B merupakan array *byte* kunci dan n merupakan iterasi 0≤ n≤ 255 , hasil dari array S indeks ke S[i] + S[j]. Algoritmanya seperti contoh algoritma 10 dibawah ini.

Algorithm 4 Prosedur *key-scheduling*.

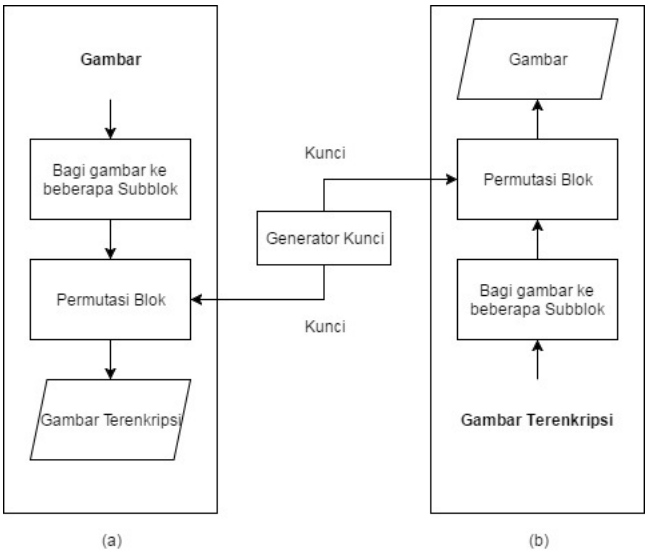
```
1: procedure KEYSCHEDULINGALGORITMA
2:     j = 0
3:     For j = 0 : 255 do                          ▷ Pemberian nilai awal
4:         j = (j + S[i] + K[i]) mod 256
5:         Swap (S[i],S[j])
6:     EndFor
7: end procedure
```

Setelah mendapatkan *byte* kunci, nanti akan di-XOR-kan dengan *byte* pesan atau data yang akan disisipkan.

3.1 Algortima Permutasi Gambar

Permutasi merupakan salah satu proses enkripsi pada gambar dengan cara membagi dan mengganti susunan pada gambar asli.Terdapat 3 basic permutasi [2] yang dapat dilakukan pada gambar yaitu,

- Permutasi bit Dalam teknik permutasi bit, tiap bit pada gambar dipilih dan dipermutasi dengan kunci yang dipilih dari sekumpulan kunci dengan menggunakan *pseudo random index generator*.



Gambar 3.1: Blok Diagram *Permutasi* (a)Enkripsi (b)Dekripsi

- Permutasi pixel Dalam teknik permutasi pixel, pixel diambil dari gambar dandipermutasi menggunakan kunci yang dipilih dari kumpulan kunci. Permutasi pixel dapat diterapkan dengan cara yang berbeda-beda tergantung ukuran kuncinya. Jika kunci berukuran satu dimensi maka dapat diterapkan dengan melakukan permutasi baris dan kolom sesuai dengan kuncinya dan jika ukuran kunci dua dimensi maka pixel diletakkan sesuai dengan posisi pada kunci.

- Permutasi blok Blok permutasi diterapkan dengan membagi gambar menjadi beberapa subblok. Proses permutasi blok sama dengan permutasi pixel.

Pada tugas akhir ini mengimplementasikan permutasi blok. Blok diagram dari permutasi dapat dilihat pada 3.1. Gambar dibagi menjadi N subblok.

Generator kunci digunakan untuk menghasilkan satu dimensi kunci ukuran N dengan rentang nilai dari "0" sampai dengan "N-1". Lalu tiap subblok

akan dipermutasi berdasarkan kunci yang dihasilkan dengan menukar posisinya seperti algoritma 5. Pembangkitan kunci dapat dilakukan dengan fungsi LCG dan lain-lain.

Algorithm 5 Prosedur Permutasi Blok.

```
procedure PERMUTASI
    Start
    Bangkitkan Kunci sebanyak N kunci (K)
    For i = 0 : N − 1 do                          ▷ Pemberian nilai awal
        Tukar (B[i] dengan B[K[i]])
    EndFor
    End
end procedure
```

3.2 Pengukuran Kualitas Citra

Analisi kualitas citra dilakukan dengan pendekatan *fidelity*, yaitu pengujian tarehadap aspek mutu. Beberapa metode yang digunakan untuk mengukur aspek mutu suatu citra, yaitu

3.2.1 *Mean Square Error* (MSE)

MSE meruakan kesalahan kuadrat kumulatif antara stego dan stego cover yang dinyatakan ke dalam bentuk persamaan 3.1

$$MSE = \frac{1}{XY}\sum_{y=1}^{Y}\sum_{x=1}^{X}[Ix, y - K(x, y)]^2 \tag{3.1}$$

dimana *i(x,y)* dan *K(x,y)* merupakan nilai pixel pada posisi x,y di cover stego atau mewakili dimensi gambar.

3.2.2 *Peak Signal Noise Ratio* (PSNR)

PSNR merupakan suatu ukuran variasi kualitas antara biner terakhir dan cover stego

$$PSNR = 20log_{10}(\frac{MAX_1}{\sqrt{MSE}}) \tag{3.2}$$

dimana MAX_1 adalah nilai piksel maksimum. Semakin besar nilai PSNR maka semakin sedikit perbedaan biner terakhir dengan cover stego.

Bab IV

Metodologi dan Desain Sistem

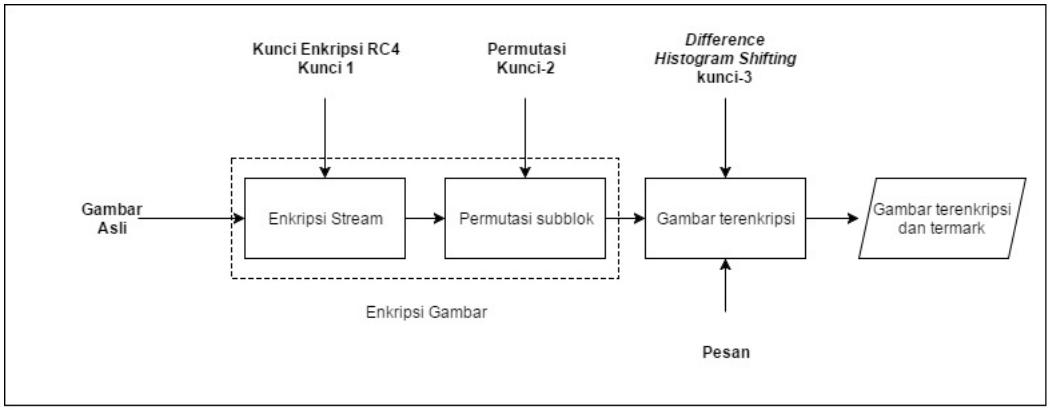
4.1 Gambaran Umum Sistem

Pembangunan sistem yang dapat melakukan penyisipan pesan berupa riwayat penyakit pasien ke dalam gambar X-ray atau HasilCT scan pasien yang terenkripsi merupakan penelitian yang akan dilakukan pada tugas akhir ini. Tujuan dari sistem ini adalah agar pesan dapat disisipkan pada gambar pasien yang terenkripsi dan dapat mengembalikan gamabar seperti semula setelah pesan dienkripsi.

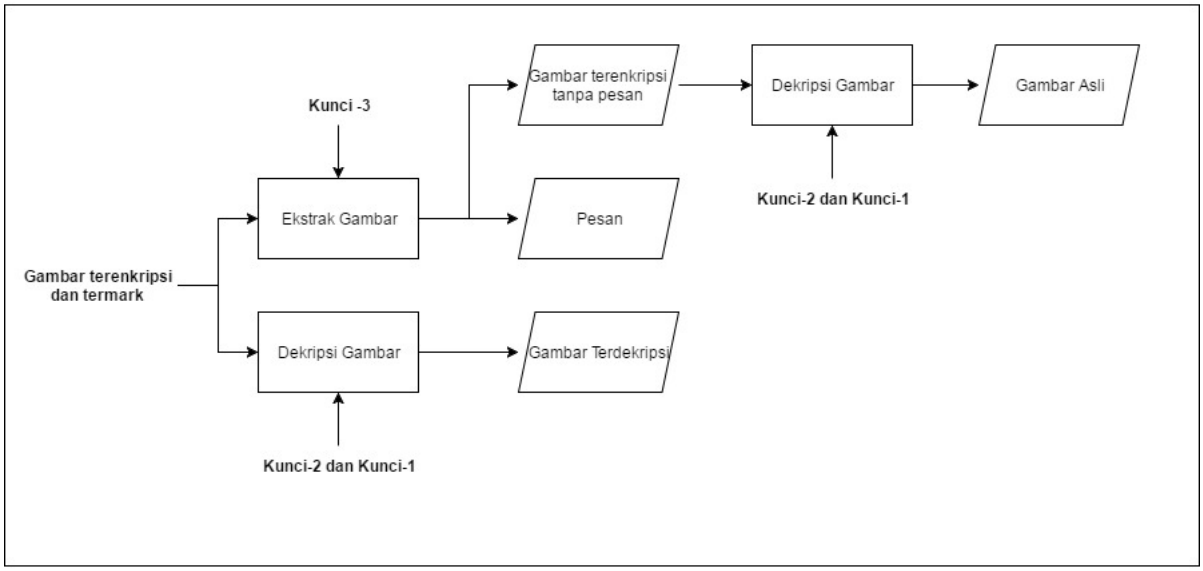
Sistem ini terdiri dari dua bagian, yang pertama adalah enkripsi gambar dan penyisipan gambar seperti yang tergambar pada gambar 4.1(a). Lalu yang kedua adalah bagian ekstraksi data dan pemulihan gambar seprti yang tergambar pada gambar 4.1(b).

Pada sisi pegirim, media gambar akan di bagi menjadi beberapa subblok. Lalu, dengan menggunakan stream chiper RC4 gambar yang telah dibagi-bagi akan enkripsi menggunakan kunci enkripsi kunci-1. setelah itu, tiap subblok dari gambar yang telah terenkripsi dipermutasi menggunakan kunci permutasi kunci-2. Pesan akan disisipkan kedalam gambar secara *reversible* menggunakan skema *difference histogram shifting*.

Pada Sisi penerima, terdapat 2 kasus. Kasus pertama adalah penerima hanya akan mendekripsi gambar tanpa ekstraksi data menggunakan kunci-1 dan kunci-2. Kasus kedua adalah penerima mengdekripsi gambar dan memulihkan gambar seperti aslinya.



(a)



(b)

Gambar 4.1: Blok Diagram *reversible data hiding* (a)Enkripsi Gambar dan Penyisipan Data (b)Ekstraksi Data dan Peulihan Gambar

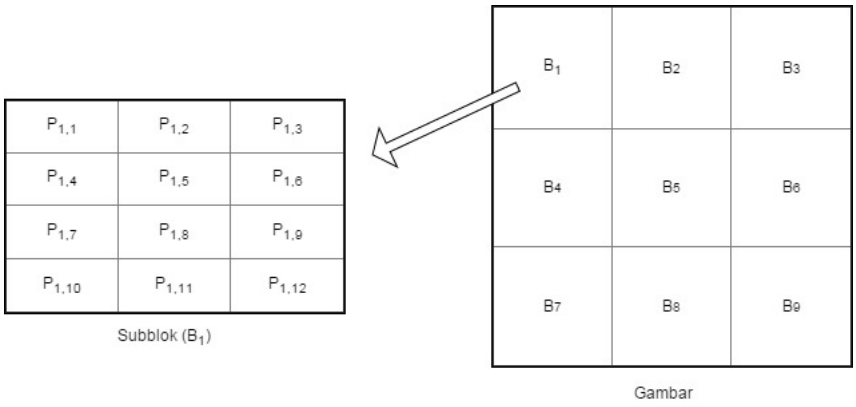
4.2 Penjelasan Rancangan Sistem

Pada bagian ini akan dijelaskan secara rinci dari blok diagram 4.1. Gambar yang digunakan pada penelitian ini berupa medis hasil *CT scan* (X-ray) sepert gambar 4.2.

Gambar 4.2: Contoh Gambar

4.2.1 Enkripsi Gambar

Enkripsi gambar diawali dengan melakukan pembagian gambar I menjadi N buah subblok (B_1,B_2,..,B_N) takberimpitan satu sama lainnya yang berukuran m x n. Sehingga dapat direpresentasikan pixel pada subblok B_i dengan $P_{i,j}(1 \leq i \leq N,1 \leq j \leq m\ x\ n)$ dimana i mengrepresentasikan indeks dari subblok dan j adalah



Gambar 4.3: Representasi Subblok

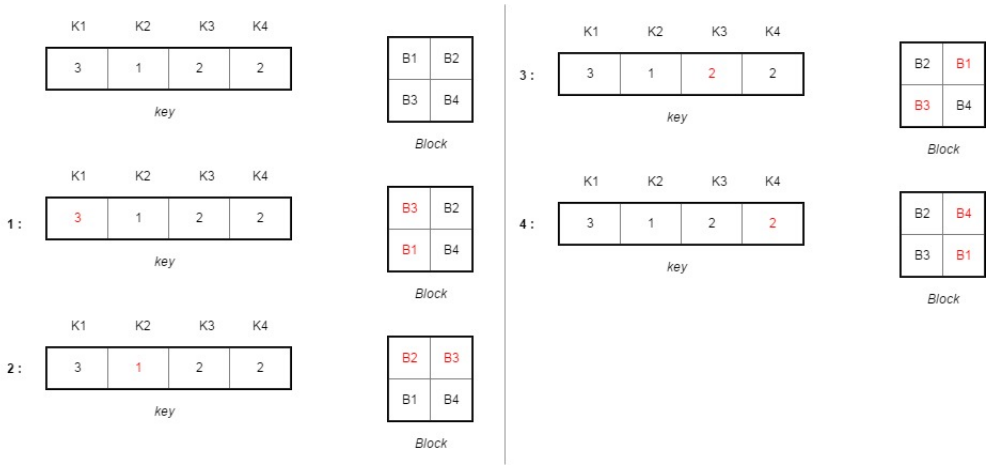
Berdasarkan kunci-1 jalankan *stream chiper* RC4 untuk menghasilkan kunci *stream* sepanjang N *byte*. Kunci *stream* direpresentasikan dengan $R_i(1 \leq i \leq N)$ dimana i merupakan indeks dari kunci *stream* yang panjangnya 8 bit.

Untuk melakukan enkripsi pada gambar, lakukan operasi (XOR) antara $P_{i,j}$ dengan R_i, seperti persamaan 4.1. Setelah itu gambar diermutasi antar

subblok menggunakan kunci permutasi kunci-2.

$$E_{i,j} = P_{i,j} \oplus R_i(1 \leq j \leq m \ x \ n) \tag{4.1}$$

Setelah itu permutasi semua semua subblok dengan kunci permutasi kunci-2. Sebagai catatan pada tahap ini hanya urutan dari tiap subblok yang akan dipermutasi sedangakan urutan pixel pada tiap subblok tetap sama. Proses permutasi dilakukan mengikuti algoritma 5 dan blok diagram permutasi 3.1.Langkah pertama adalah menbangkitkan kunci dari bilangan random berukuran N dengan rentang nilai tiap kunci dari 1 sampai dengan N (K_1,K_2,..,B_N). Setelah itu tiap subblok (B_1,B_2,..,B_N) akan dipermutasi dengan menukar posisi tiap subblok sesuai dengan indeks kunci. Untuk memudahkan, diilustrasikan pada gambar 4.4.



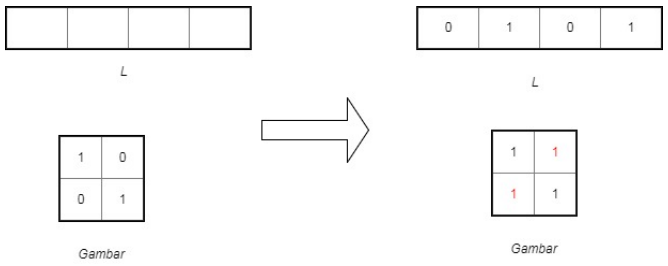
Gambar 4.4: Ilustrasi Permutasi Blok

4.2.2 Penyisipan Data

Penyisipan data pada gambar terenkripsi (gambar)menggunakan metode *difference histogram shifting*.Langkah pertama dari metode ini adalah menginisialisasi lokasi map L kekadaan kosong. Kunjungi tiap2 pixel dan masukkan "0" ke L ketika $C_{i,j} \subset [1, 254]$ dan "1" jika $C_{i,j} \subset [0, 255]$ dan modifikasi $C_{i,j}$ ke $C'_{i,j}$ menggunakan persamaan 4.2,

$$C'_{i,j} = \begin{cases} 254, & jika\ C_{i,j} = 255 \\ 1, & jika\ C_{i,j} = 0 \\ C_{i,j}, & lainnya \end{cases} \tag{4.2}$$

ilustrasi dapat dilihat seperti gambar 4.5



Gambar 4.5: Ilustrasi *Location Map*

Lalu dilanjut dengan mengmodifikasi histogram dengan melihat nilai selisih dari tiap blok megikuti persamaan 4.3

$$D_{ij} = C_{i,j} - C_{i,1} \tag{4.3}$$

berdasarkan analisis pada literatur [3], nilai puncak dari *difference histogram* akan berada pada poin 0, 1 dan -1. sehingga algoritma penyisipan skema RDH dapat mengikuti persaaan 4.4,

$$C''_{i,j} = \begin{cases} C'_{i,j} - 1, & jika\ D_{i,j} < -1 \\ C'_{i,j} - b, & jika\ D_{i,j} = -1 \\ C'_{i,j} + b, & jika\ D_{i,j} = 0 \\ C'_{i,j} + 1, & jika\ D_{i,j} > 0 \end{cases} \tag{4.4}$$

dimana $b \subset [0,1]$ merupakan pesan yang disisipkan.Dari persamaan 4.4 pesan akan disisipkan kedalam gambar terenkripsi jika nilai dari selisih $D_{i,j}$ bernilai sama dengan "-1" atau "0".

4.2.3 Dekripsi Gambar dan Ekstraksi Data

Proses ekstraksi dan pemulihan gambar dapat dijelaskan seperti persamaan 4.5 dan 4.6 .

$$b^* = \begin{cases} 0, & jika\ C''_{i,j} - \ C''_{i,1} = 0, -1 \\ 1, & jika\ C''_{i,j} - \ C''_{i,j} = 1, -2 \end{cases} \tag{4.5}$$

dimana b^* mengrepresentasikan bit pesan yang diekstrak. Pada proses dapat kita lihat dari persamaan 4.5 b^* bernilai "0" jika selisih dari $C''_{i,j}$ - $C''_{i,1}$ sama dengan "0" atau "1", dan bernilai "1" jika selisihnya sama dengan "1" atau "-2".

$$C'_{i,j}{}^* = \begin{cases} C''_{i,j} - 1, & jika\ C''_{i,j} - C''_{i,1} > 0 \\ C''_{i,j} + 1, & Jika\ C''_{i,j} - C''_{i,j} < -1 \\ C''_{i,j}, & lainnya \end{cases} \tag{4.6}$$

dimana $C'_{i,j}{}^*$ merepresentasikan nilai pixel yang dipulihkan setelah ekstraksi pesan. setelah pesan dan gambar direstorasi maka gambar yang terenkripsi bisa dipulihkan dengan menggunakan lokasi map L mengikuti persamaan dibawah ini,

$$C^*_{i,j} = \begin{cases} 255, & jikaC'_{i,j}{}^* = 254 \\ 0, & jikaC'_{i,j}{}^* = 1 \\ C'_{i,j}{}^*, & otherwise \end{cases} \tag{4.7}$$

4.3 Gambaran Singkat Sistem

Secara umum, sistem ini nantinya dapat digunakan pada dunia medis. Sistem ini akan terdiri dari dua sisi pengguna yaitu dokter dan pasien. Masukan pada sistem ini berupa gambar X-ray dari pasien dan data rahasia pasien berupa riwayat atau hasil diagnosa dari gambar X-ray yang disisipkan kedalam gambar X-ray oleh dokter. Pada sisi dokter, dokter dapat menyisipkan dan mengenkripsi diagnosa pasien kedalam gambar X-ray. Gambar yang telah tersisipkan pesan dan dienkripsi dikirim kepasien atau disimpan kedalam database. Selain itu, dokter dapat mengdekripsi gambar dan dan mengekstrak pesan yang tersisipkan. Berbeda dengan pasien, pasien hanya bisa mendekripsi gambar yang nantinya gambar tersebut dapat dibawa kedokter untuk dikonsultasikan penyakitnya.

Daftar Pustaka

[1] James M Barton. Method and apparatus for embedding authentication information within digital data, July 8 1997. US Patent 5,646,997.

[2] Ravi Prakash Dewangan and Chandrashekhar Kamargaonkar. Image encryption using random permutation by different key size. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 2015.

[3] Fangjun Huang, Jiwu Huang, and Yun-Qing Shi. New framework for reversible data hiding in encrypted domain. *IEEE Transactions on Information Forensics and Security*, 11(12):2777–2789, 2016.

[4] Rhythm Katira and V Thanikaiselvan. Random traversing based reversible data hiding technique using pe and lsb. *IJET: International Journal of Engineering and Technology*, 2013.

[5] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su. Reversible data hiding. *IEEE Transactions on circuits and systems for video technology*, 16(3):354–362, 2006.

[6] Jun Tian. Reversible data embedding using a difference expansion. *IEEE transactions on circuits and systems for video technology*, 13(8):890–896, 2003.

Lampiran

Bab V

Algoritma

Algorithm 6 Prosedur inisialisasi *state-array*.

1: **procedure** MainEncryption
2: Start
3: Read $Image$
4: S, K
5: definitionBlok($defBlok$,$Image$) ▷ Algoritma 7
6: inisialisasiStateArray(S, K) ▷ Algoritma 8
7: KeySchedulingAlgoritma(S) ▷ Algoritma 9
8: PRGA($defBlok$,$Image$) ▷ Algoritma 10
9:
10: permutationBlok($Image$,$defBlok$) ▷ Algoritma 13
11:
12: shifftHistogram($L, Image$) ▷ Algoritma 17
13: insertMassage($Image, M, defBlok$) ▷ Algoritma 18
14: End
15: **end procedure**

Algorithm 7 Prosedur xor blok.

```
1: procedure definitionBlok(in/out defBlok,in Image)
2:     Start
3:     i = 0
4:     j = 0
5:     While i <lebar dari gambar do                ▷ Pemberian nilai awal
6:         While j <panjang dari gambar do          ▷ Pemberian nilai awal
7:             pos.x= i
8:             pos.y= j
9:             defBlok.append(pos)
10:        EndWhile
11:    EndWhile
12:    End
13: end procedure
```

Algorithm 8 Prosedur inisialisasi *state-array*.

```
1: procedure inisialisasiStateArray(in/out S,in/out K)
2:     Start
3:     For i = 0 : 255 do                           ▷ Pemberian nilai awal
4:         S[i] = i
5:         K[i] = key[i mod key.length()]
6:     EndFor
7: end procedure
```

Algorithm 9 Prosedur *key-scheduling*.

```
1: procedure KeySchedulingAlgoritma(in/out S)
2:     j = 0
3:     For j = 0 : 255 do                           ▷ Pemberian nilai awal
4:         j = (j + S[i] + K[i]) mod 256
5:         Swap (S[i],S[j])                         ▷ Algoritma 11
6:     EndFor
7: end procedure
```

Algorithm 10 Prosedur Pembangkitan Kunci Enkripsi.

```
1: procedure PRGA(in/out defBlok,in/out img)
2:     Start
3:     i = 0
4:     j = 0
5:     For n = 0 : defBlok.size do                    ▷ Pemberian nilai awal
6:         i= (i + 1) mod 256
7:         j= (j + S[i]) mod 256
8:         Swap (S[i], S[j])                          ▷ Algoritma 11
9:         B= s[S[i] + S[j]] mod 256
10:        xorBlok(B, defBlok[n], img)                ▷ Algoritma 12
11:    EndFor
12:    End
13: end procedure
```

Algorithm 11 Prosedur penukaran Nilai.

```
1: procedure SWAP(in/out val1,in/out val2)
2:     Start
3:     temp = val1
4:     val1 = val2
5:     val1 = temp
6: end procedure
```

Algorithm 12 Prosedur xor blok.

```
1: procedure XORBLOK(in B,in/out pos,in/out img)
2:     Start
3:     For i = 0 : 2 do                               ▷ Pemberian nilai awal
4:         For j = 0 : 2 do                           ▷ Pemberian nilai awal
5:             img[pos.x + i][pos.y + j]= img[pos.x + i][pos.y + j] xor B
6:         EndFor
7:     EndFor
8:     End
9: end procedure
```

Algorithm 13 Prosedur Permutasi Blok

```
1: procedure permutationBlok(in/out img,in defBlok)
2:     Start
3:     p= bilanganPrima(256)                          ▷ Algoritma 14
4:     a= generateNumber(p)
5:     For i = 0 :size of defblok do                  ▷ Pemberian nilai awal
6:         d = pangkat(a, i + 69mod256)mod256)
7:         SwapBlok(defBlok[i], defBlok[d − 1], img)  ▷ Algoritma 16
8:     EndFor
9: end procedure
```

.

Algorithm 14 Prosedur Generate bilangan Prima.

```
1: procedure bilanganPrima(in N)
2:     Start
3:     For i = N :N + 100 do
4:         If isPrima(N) Then                         ▷ Algoritma 15
5:             return N
6:     EndFor
7:     End
8: end procedure
```

Algorithm 15 Prosedur cek Bilangan Prima.

```
1: procedure IsPrima(in N)
2:     Start
3:     For j = 2 :akar(N) + 1 do
4:         If Nmod2 == 0 Then
5:             return false
6:     EndFor
7:     End
8: end procedure
```

Algorithm 16 Prosedur Pertukaran Blok.

```
1: procedure SwapBlok(in b1,in b2,in/out img)
2:     Start
3:     For i = 0 : 2 do
4:         For j = 0 : 2 do
5:             temp = img[b1.x + i][b1.y + j]
6:             img[b1.x + i][b1.y + j] = img[b2.x + i][b2.y + j]
7:             img[b2.x + i][b2.y + j] = temp
8:         EndFor
9:     EndFor
10:    End
11: end procedure
```

Algorithm 17 Prosedur shiffting.

```
1: procedure shiftHistogram(in L,in/out img)
2:     Start
3:     For i = 0 :lebar img do
4:         For j = 0 :panjang img do
5:             If img[i][j] == 1 or img[i][j] == 254 Then
6:                 L.append(0)
7:             Else If img[i][j] == 255 Then
8:                 L.append(1)
9:                 img[i][j] = 1
10:            Else If img[i][j] == 255 Then
11:                L.append(1)
12:                img[i][j] = 254
13:        EndFor
14:    EndFor
15:    End
16: end procedure
```

Algorithm 18 Prosedur shiffting.

```
1: procedure insertMassage(in/out img,in M,in defBlok)
2:     Start
3:     n = 0
4:     For b = 0 :Ukuran defBlok do
5:         For i = defBlok[b].y : defBlok[b].y + 2 do
6:             For j = defBlok[b].x : defBlok[b].x + 2 do
7:                 D = img[i][j] − img[i][1]
8:                 If D < −1 Then
9:                     img[i][j]img[i][j] - 1
10:                Else If D == 1 Then
11:                    img[i][j]img[i][j] - M[n]
12:                    n = n + 1
13:                Else If D == 0 Then
14:                    img[i][j]img[i][j] + M[n]
15:                    n = n + 1
16:                Else If D > 0 Then
17:                    img[i][j]img[i][j] + 1
18:                EndFor
19:            EndFor
20:        EndFor
21:    EndFor
22:    End
23: end procedure
```

Algorithm 19 Prosedur inisialisasi *state-array*.

```
1: procedure inisialisasiStateArray
2:     Start
3:     For i = 0 : 255 do                          ▷ Pemberian nilai awal
4:         S[i] = i
5:         K[i] = key[i mod key.length()]
6:     EndFor
7: end procedure
```

Algorithm 20 Prosedur *key-scheduling*.

```
1: procedure KeySchedulingAlgoritma
2:     j = 0
3:     For j = 0 : 255 do                          ▷ Pemberian nilai awal
4:         j = (j + S[i] + K[i] + 23) mod 256
5:         Swap (S[i],S[j])                        ▷ Algoritma 11
6:     EndFor
7: end procedure
```

Algorithm 21 Prosedur Pembangkitan Kunci Enkripsi.

```
1: procedure PRGA
2:     Start
3:     i = 0
4:     j = 0
5:     For n = 0 : PlainSize do                    ▷ Pemberian nilai awal
6:         i= (i + 1) mod 256
7:         j= (j + S[i]) mod 256
8:         Swap (S[i], S[j])                       ▷ Algoritma 11
9:         B= s[S[i] + S[j]] mod 256
10:        chipper = BxorChiperText                ▷ Algoritma 12
11:    EndFor
12:    End
13: end procedure
```