

# PROGRAM APLIKASI KEAMANAN CITRA DENGAN ALGORITMA *DES* DAN TRANSFORMASI *WAVELET* DISKRIT

Solichin Zaki<sup>1</sup>, Bayu Surarso<sup>1</sup> dan Eko Adi Sarwoko<sup>2</sup>

<sup>1</sup>Jurusan Matematika FMIPA UNDIP

<sup>2</sup>Jurusan Teknik Informatika FMIPA UNDIP

Jl. Prof. H. Soedarto, S. H, Tembalang, Semarang

**Abstract.** Application program security image using DES cryptographic algorithm and discrete wavelet transform is an application program that is used to secure image transmission through the Internet. Manufacturing processes are as follows, first is to make the program application for image encryption ciphertext is a program that is used to transform the initial image into the form of a compressed image using advanced discrete wavelet transform (DWT) and the Haar filter, then encrypt image transformation results with using the DES encryption algorithm specified by the encryption key, the encrypted form of information in the form of ciphertext. The second process is the process for making an application program to restore the initial image into the form ciphertext, the ciphertext decrypt the encrypted image and the particular key (the key with encryption) using DES decryption algorithm, the decrypted image, then decrypted form of the image is brought back into form initial image using inverse discrete wavelet transformation (IDWT) and reconstruction program.

**Keywords:** Image, Cryptography, Wavelet, DES, DWT, IDWT

## 1. PENDAHULUAN

Salah satu komponen multimedia yang berperan sangat penting dalam bentuk informasi visual adalah Gambar atau Citra. Informasi yang berbentuk citra mempunyai karakteristik yang berbeda dengan teks dan citra memberikan informasi yang lebih banyak dibanding dengan informasi berbentuk teks [2]. Pada umumnya untuk melakukan pengkodean suatu citra, menentukan domain yang lebih sesuai untuk proses pengkuantisasian harus melakukan transformasi. Metode yang banyak digunakan dalam transformasi ini antara lain Transformasi *Cosinus diskrit*, Transformasi *Fourier* dan Transformasi *Wavelet*. Dari ketiga jenis transformasi tersebut, transformasi *wavelet* dianggap paling baik hasilnya [1].

Perkembangan informasi melalui jaringan internet membuat pertukaran informasi semakin cepat dan akurat serta terbuka melewati batas-batas negara dan budaya. Perkembangan ini akan menimbulkan tidak hanya dampak positif yang menguntungkan bagi dunia komunikasi dan pertukaran informasi saja tetapi juga berdampak negative yaitu

kejahatan komputer antara lain pencurian, penipuan pemerasan dan lainnya.

Keamanan informasi pada komputer tidak hanya bergantung pada *firewall* dan deteksi sistem intruksi saja tetapi juga keamanan informasi dari informasi itu sendiri. Kriptografi memegang peran penting dalam membangun keamanan informasi. Kriptografi bertujuan agar pesan informasi tidak dapat dibaca oleh orang yang tidak berhak sehingga informasi baik yang disimpan dalam computer aman maupun yang dikirim melalui jaringan komputer aman dan bisa dipertanggung jawabkan oleh sipengirim. *DES* merupakan algoritma standar yang sampai saat ini masih banyak digunakan dan masih dianggap aman untuk menjawab tantangan perkembangan teknologi komunikasi yang sangat cepat [2].

Dari hal-hal tersebut diatas, timbul masalah-masalah. Dengan transformasi yang bagaimana membuat aplikasi sistem keamanan agar informasi yang berbentuk citra dapat disimpan dengan aman dan informasi yang dikirim melalui jaringan komunikasi internet sampai pada tujuan yang berhak dengan aman sehingga dapat dipertanggung jawabkan.

Permasalahan sistem keamanan informasi pada penelitian ini, hanya difokuskan pada pembahasan algoritma *DES* dan aplikasi transformasi *wavelet* diskrit yang digunakan untuk keamanan informasi berbentuk citra, dengan tujuan:

- a. Melakukan teknik pemrosesan sinyal citra digital dengan menggunakan transformasi *wavelet* diskrit dengan *filter Haar*.
- b. Melakukan enkripsi dan dekripsi sinyal citra digital dengan algoritma *DES*.
- c. Membuat program aplikasi keamanan citra dengan algoritma *DES*.
- d. Menganalisis program aplikasi keamanan citra dengan algoritma *DES*.

## 2. TEORI

### 2.1 Transformasi *Wavelet* Diskrit

Transformasi *Wavelet* adalah sebuah fungsi variabel riil  $t$  yang digunakan untuk melokalisasi suatu fungsi dalam ruang dan skala  $L2(R)$ , diberi notasi  $\psi(t)$  sebagai *mother wavelet*. *Wavelet*  $\psi_{a,b}(t)$  dihasilkan oleh parameter dilatasi  $a$  dan translasi/kontraksi  $b$ , yang dinyatakan dalam persamaan :

$$\psi_{a,b}(t) = a^{-1/2} \psi\left(\frac{t-b}{a}\right); \text{ dengan } a > 0, b$$

bilangan-bilangan integer

dengan :  $a$  = parameter dilatasi atau kontraksi,  $b$  = parameter translasi

Formula *Calderon*

$$(f)(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt,$$

$$f(t) = C \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \langle f, \psi_{a,b} \rangle \psi_{a,b}(t) a^{-2} da db.$$

Fungsi *Haar* :

$$\psi(t) = \begin{cases} 1 & , 0 \leq t \leq \frac{1}{2} \\ -1 & , \frac{1}{2} \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

dan  $\psi_{j,k}(t) = a^{j/2} \psi(2^j t - k)$ ;  $j, k \in Z(\text{integer})$  dengan:  $j$  integer nonnegative,  $0 \leq k \leq 2^{j-1}$ ,  $2^j$  = parameter dilatasi

(parameter frekuensi atau skala),  $k$  = parameter waktu atau lokasi ruang dan memenuhi kondisi  $\int_{-\infty}^{\infty} \psi(t) dt = 0$  [1].

Berdasarkan nilai parameter translasi dan dilatasinya, transformasi *wavelet* dibedakan menjadi dua tipe, yaitu *Continue Wavelet Transform (CWT)* dan *Discrete Wavelet Transform (DWT)*.

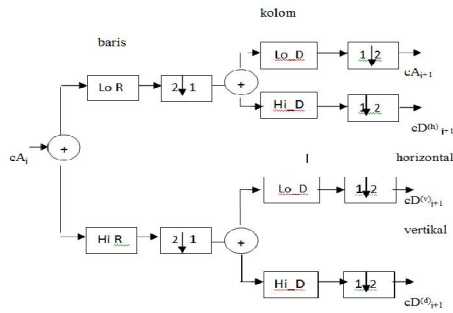
a. Transformasi *wavelet* kontinu ditentukan oleh nilai parameter dilatasi ( $a$ ) dan translasi ( $b$ ) yang bervariasi secara kontinu, dimana  $a, b \in R$  dan  $a \neq 0$ . *Continue Wavelet Transform (CWT)* menganalisis sinyal dengan perubahan skala pada *window* yang dianalisis, pergeseran *window* dalam waktu dan perkalian sinyal serta mengintegral semuanya sepanjang waktu. Secara matematis dirumuskan sebagai :

$$CWT(a,b) = \int f(t) \psi_{a,b}^*(t) dt$$

b. Transformasi *wavelet* diskrit bertujuan untuk mengurangi redundansi yang terjadi pada transformasi *wavelet* kontinu dengan cara mengambil nilai diskrit dari parameter  $a$  dan  $b$ , serta menganalisis suatu sinyal dengan skala yang berbeda dan merepresentasikannya ke dalam skala waktu dengan teknik *filtering*, yaitu melewati sinyal dalam domain waktu ke dalam *High Pass Filter* dan *Low Pass Filter* dan untuk memisahkan komponen frekuensi tinggi dan frekuensi rendah dengan *filter* yang berbeda frekuensi *cut-off*-nya. Transformasi *Wavelet* Diskrit (*DWT*) dikelompokkan menjadi dua yaitu *DWT* maju dan *DWT* balik [1].

### Transformasi *Wavelet* Diskrit Maju (*Forward DWT*)

*DWT* maju merupakan proses dekomposisi data citra, yang dimulai dengan melakukan dekomposisi terhadap baris data citra dan dekomposisi terhadap kolom pada koefisien citra keluaran dari tahap pertama[7]. Proses *DWT* maju, ditunjukkan pada Gambar 1 :



**Gambar 1** Forward DWT Dua Dimensi Skala Satu

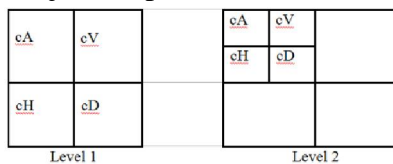
Keterangan Gambar 1:

$cA_j$  = citra masukan,

$\begin{matrix} \downarrow 2 \\ \hline \end{matrix}$  = down sampling baris ,  
 $\begin{matrix} \downarrow 1 \\ \hline \end{matrix}$  = down sampling kolom

$cA_{j+1}$  = koefisien aproksimasi ,  
 $cD^{(h)}_{j+1}$  = koefisien detail horizontal,  
 $cD^{(v)}_{j+1}$  = koefisien detail vertical,  
 $cD^{(d)}_{j+1}$  = koefisien detail diagonal

Sinyal citra masuk, didekomposisi menggunakan *Lo\_D* (Low Pass Filter Decomposition) dan *Hi\_D* (High Pass Filter Decomposition) dan dilakukan downsampling dua. Sinyal keluaran berfrekuensi rendah dan tinggi. Proses *Lo\_D* dan *Hi\_D* tersebut dilakukan terhadap baris dan terhadap kolom sebanyak dua kali, diperoleh empat subband keluaran berupa informasi frekuensi rendah dan informasi frekuensi tinggi, yaitu koefisien aproksimasi, koefisien detail horizontal, koefisien detail vertikal, dan koefisien detail diagonal. Dekomposisi transformasi wavelet [5], ditunjukkan pada Gambar 2 :

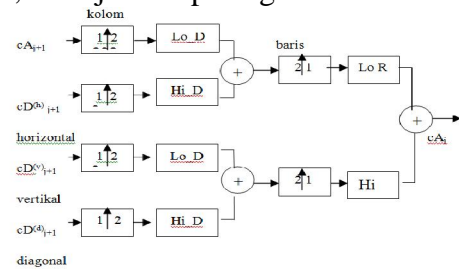


**Gambar 2** Transformasi wavelet level 1 dan level 2

**Transformasi Wavelet Diskrit Balik (Invers DWT)**

DWT balik (IDWT) adalah kebalikan dari DWT maju, yaitu proses rekonstruksi dengan arah yang berlawanan dari proses dekomposisi. Proses up-sampling dan

pem-filter-an dengan koefisien-koefisien filter balik. Proses up-sampling yaitu proses mengembalikan dan menggabungkan kesinyal semula, yaitu dengan menyisipkan sebuah kolom nol di antara setiap kolom dan melakukan konvolusi pada setiap baris dengan filter berdimensi satu begitu juga menyisipkan sebuah baris nol di antara setiap baris dan melakukan konvolusi setiap baris dengan filter yang lainnya. Filter yang digunakan pada transformasi balik harus sesuai dengan filter pada sisi dekomposisi yaitu filter *Lo\_R* (Low Pass Filter Reconstruction) dan *Hi\_R* (High Pass Filter Reconstruction)[8]. Proses DWT balik, ditunjukkan pada gambar 2.3:



**Gambar 3** Backward DWT Dua Dimensi Skala Satu

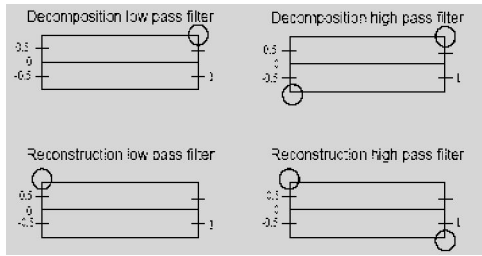
Keterangan Gambar 3:

$cA_j$  = citra keluaran (sama dengan citra masukan)

$\begin{matrix} \uparrow 2 \\ \hline \end{matrix}$  = up sampling baris ,  
 $\begin{matrix} \uparrow 1 \\ \hline \end{matrix}$  = up sampling kolom  
 $\begin{matrix} \uparrow 1 \\ \hline \end{matrix}$  = koefisien aproksimasi,  
 $\begin{matrix} \uparrow 2 \\ \hline \end{matrix}$  = koefisien detail horizontal,  
 $cD^{(v)}_{j+1}$  = koefisien detail vertical,  
 $cD^{(d)}_{j+1}$  = koefisien detail diagonal

**Pemilihan Filter Wavelet**

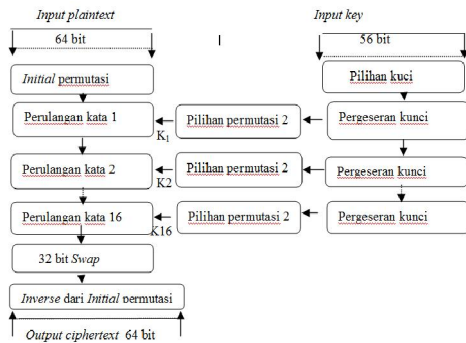
Pada tulisan ini, digunakan wavelet dengan filter Haar (Daubechies orde 1). Wavelet dengan filter Haar dipilih karena memiliki low pass filter dan high pass filter yang tidak memakan biaya komputasi yang besar [6]. Filter Haar, ditunjukkan pada Gambar 4.



Gambar 4 Filter Haar

## 2.2 Algoritma Enkripsi DES

DES termasuk dalam algoritma enkripsi yang sifatnya *cipher block*, yaitu mengubah data masukan menjadi blok-blok 64-bit dan menggunakan kunci enkripsi sebesar 56-bit. Proses enkripsi menghasilkan output blok 64-bit. Secara umum algoritma enkripsi DES [4], ditunjukkan pada gambar.



Gambar 5 Algoritma Enkripsi DES

## 2.3 Algoritma Dekripsi DES

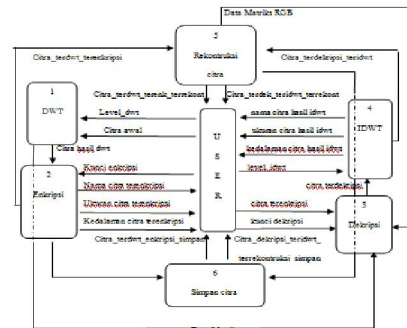
Proses dekripsi Algoritma DES, pada intinya sama seperti pada proses enkripsi, prosesnya balikan dari proses enkripsi. Blok  $(R_{16}, L_{16})$  merupakan masukan awal untuk *deciphering*. Blok  $(R_{16}, L_{16})$  diperoleh dengan mempermutasikan *ciphertext* dengan matriks permutasi  $IP^{-1}$ . Prakeluaran dari *deciphering* adalah  $(R_0, L_0)$ . Dengan permutasi awal IP akan didapatkan kembali blok *plaintext* semula.  $K_{16}$  dihasilkan dari  $(C_{16}, D_{16})$  dengan permutasi  $PC-2$ . Tentu saja  $(C_{16}, D_{16})$  tidak dapat diperoleh langsung pada permulaan *deciphering*. Tetapi karena  $(C_{16}, D_{16}) = (C_0, D_0)$ , maka  $K_{16}$  dapat dihasilkan dari  $(C_0, D_0)$  tanpa perlu lagi melakukan pergeseran bit.  $(C_0, D_0)$  merupakan bit-bit dari kunci eksternal K yang diberikan pada

waktu dekripsi. Selanjutnya  $K_{15}$  dihasilkan dari  $(C_{15}, D_{15})$  yang diperoleh dengan menggeser  $(C_{16}, D_{16}) = (C_0, D_0)$  satu bit kekanan (sesuai arah kebalikan dari tabel *schedule of left*).  $K_{14}$  dihasilkan dari  $(C_{14}, D_{14})$  yang diperoleh dengan menggeser  $(C_{15}, D_{15})$  dua bit kekanan (sesuai arah kebalikan dari tabel *schedule of left*), dan seterusnya sampai dengan  $K_1$  yang dihasilkan dari  $(C_1, D_1)$ . Secara umum  $(C_{i-1}, D_{i-1})$  diperoleh dengan menggeser  $(C_i, D_i)$  satu atau dua bit kekanan (sesuai arah kebalikan dari tabel *schedule of left*) [3].

## 3. METODE PENELITIAN

### 3.1 Diagram Alir Data Program Aplikasi Keamanan Citra

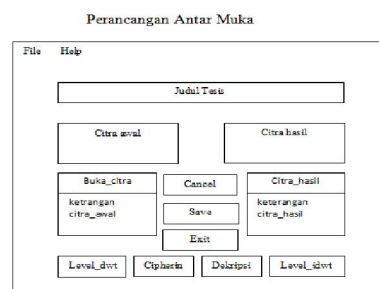
Gambaran umum dari diagram alir data program aplikasi keamanan citra dengan lingkungannya, ditunjukkan pada Gambar 6.



Gambar 6 Diagram Alur Data Program Aplikasi Keamanan Citra

### 3.2 Perancangan Bentuk Antar Muka

Bentuk utama rancangan antar muka yang akan digunakan dalam program aplikasi keamanan citra, ditunjukkan pada Gambar 7.



Gambar 7. Gambar Bentuk Utama Antar Muka

### 3.3 Proses Pembuatan Program Untuk Program Aplikasi

Didalam pembuatan program untuk program aplikasi keamanan citra ini dibuat dengan program aplikasi Matlab7.1, berikut ini adalah proses-proses yang akan dilakukan.

#### a. Proses DWT

Ambil dan baca citra yang akan diproses berupa matriks pecahan, namakan matriks image tersebut sesuai dengan  $m\_gry$  untuk gray dan untuk citra berwarna ( $R,G,B$ ):  $m\_r$  untuk red,  $m\_g$  untuk green dan  $m\_b$  untuk blue. Khusus untuk matriks  $RGB$  cukup diambil salah satu warna saja, matriks yang lain disimpan untuk proses rekonstruksi nanti. Melakukan proses transformasi  $DWT$  berdasarkan level sebagai berikut:

Untuk level\_1

untuk gray:

```
[cA1,cH1,cV1,cD1]=dwt2(m_gry,'haar');
```

```
xdwt=[cA1];
```

untuk RGB:

```
[cA1,cH1,cV1,cD1]=dwt2(m_r, 'haar');
```

```
xrdwt=[cA1];
```

Untuk Level\_2

untuk gray:

```
[cA1,cH1,cV1,cD1]= dwt2 (m_gry, 'haar');
```

```
[cA2,cH2,cV2,cD2]= dwt2 (cA1, 'haar');
```

```
xdwt=[cA2];
```

untuk RGB:

```
[cA1,cH1,cV1,cD1]= dwt2 (m_r, 'haar');
```

```
[cA2,cH2,cV2,cD2]= dwt2 (cA1, 'haar');
```

```
xrdwt=[cA2];
```

dan seterusnya.

Hasil dari transformasi ini berupa matriks pecahan juga, yang selanjutnya sebagai inputan didalam proses enkripsi.

#### b. Proses Enkripsi

Sesuai algoritma enkripsi  $DES$ , akan dibahas langkah-langkah untuk melakukan 16 kali iterasi dari matriks yang telah di  $DWT$ . Untuk kasus RGB proses enkripsi diambil sesuai dengan matriks yang diambil pada proses  $DWT$ . Langkah sebelum melakukan iterasi:

Menentukan plaintext (matriks hasil  $DWT$ ) dan mengubah plaintext kedalam bentuk bilangan biner.

Membagi plaintext\_biner menjadi plaintext1, plaintext2, plaintext3 dan seterusnya yang masing-masing terdiri dari 64 bit.

Melakukan  $IP(plaintext(j))$  permutasi awal terhadap plaintext(j) dengan IP yang telah ditentukan.

Iterasi ke(i), dengan  $i=1,2,\dots,16$

Menentukan  $L_{i-1}$  separoh (32 bit) bagian kiri dan  $R_{i-1}$  separoh (32 bit) bagian kanan dari  $IP(plaintext(i))$

Melakukan  $E(R_{i-1})$  ekspansi permutasi terhadap  $R_{i-1}$  dengan  $E$  yang telah ditentukan

Menentukan  $K$  (8 karakter) kunci yang digunakan dan mengubahnya kedalam bilangan biner (64 bit)

Melakukan  $PC_1(K)$  permutasi choice\_1 pada  $K$

Menentukan  $C_{i-1}$  separoh (28 bit) bagian kiri dan  $D_{i-1}$  separoh (28 bit) bagian kanan  $PC_1(K)$

Menentukan  $C_i$  dan  $D_i$  yaitu dengan melakukan pergeseran kekiri sesuai dengan tabel schedule of left shifts masing-masing terhadap  $C_{i-1}$  dan  $D_{i-1}$

Menentukan  $K_i = PC_2(C_i D_i)$

Menentukan  $E(R_{i-1}) \oplus K_i$  yaitu melakukan XOR :  $E(R_{i-1})$  dengan  $K_i$

Membagi  $E(R_{i-1}) \oplus K_i$  yang terdiri dari 48 bit menjadi 8 kelompok yaitu kelp\_1, kelp\_2,.....,kelp\_8 yang masing-masing terdiri 6 bit

Melakukan substitusi masing-masing kelompok ke  $S\_Box$  yaitu kelp\_1(s) ke  $S1\_Box$ , Kelp\_2(s) ke  $S2\_Box$ , ... ..kelp\_8(s) ke  $S8\_Box$ . Substitusi:

namakan bit kelompok dengan  $b_1b_2b_3b_4b_5b_6$  pada masing-masing kelompok(s), bilangan hexa dari  $b_1b_6$  menentukan baris dan bilangan hexa dari  $b_2b_3b_4b_5$  menentukan kolom pada tabel  $S\_Box$  yang bersesuaian dengan kelompoknya, bilangan hexa baris dan kolom tabel  $S\_Box$  dirubah ke bilangan biner 4 bit untuk masing-masing kelompok, diperoleh  $B_i$  32 bit.

Menentukan  $f(R_{i-1}, K_i) = P(B_i)$ , yaitu melakukan permutasi  $P$  terhadap  $B_i$  dengan permutasi  $P$  yang telah ditentukan.

Analog: untuk plaintext yang lainnya.

Dari iterasi ke<sub>16</sub> diperoleh  $(L16(i)R16(i))$ , dengan  $L16=R15$  dan  $R16=L16\oplus P(B16)$ . Selanjutnya untuk mendapatkan chipertext: Chipertext biner= $IP^{-1}(R16(i),L16(i))$  melakukan invers IP terhadap  $(R16(i),L16(i))$  dengan invers permutasi yang telah ditentukan.

#### c. Proses Dekripsi

Dari proses enkripsi iterasi ke<sub>16</sub> diperoleh  $(L16(i)R16(i))$ , dengan  $l0(i)=L16(i)=R15(i)$ ,  $r0(i)=R16(i)=L16(i)\oplus P(B16(i))$ ,  $c0=C16$ ,  $d0=D16$  dan  $K16$ . Untuk mendapatkan kembali plaintext awal, dilakukan proses dekripsi yang merupakan kebalikan dari proses enkripsi, yaitu proses yang dimulai dari iterasi ke<sub>16</sub> ke iterasi ke<sub>1</sub> sebagai berikut: untuk ciphertext1

Iterasi ke<sub>i</sub> (kebalikan dari iterasi ke<sub>16</sub> pada proses enkripsi), dengan  $i=1,2,\dots,16$ :

Menentukan  $l0(i)=L16(i)$ ,  $r0(i)=R16(i)$ ,  $c0=C16$  dan  $d0=D16$

Menentukan  $ci$  dan  $di$ , yaitu melakukan penggeseran  $c0$  dan  $d0$  kekanan sesuai dengan balikan dari schedule of left shifts

Menentukan  $ki$  baru,  $ki=cidi(P\_C2)$  yaitu permutasi  $PC\_2$  terhadap gabungan  $ci$  dan  $di$

Melakukan ekspansi permutasi  $E(ri-1)$  dan XOR:  $E(ri-1)\oplus ki$

Membagi  $E(ri-1)\oplus ki$  menjadi 8 kelompok, yaitu  $kelp\_1(s)$ ,  $kelp\_2(s)$ , ..., dan  $kelp\_8(s)$

Melakukan substitusi ke  $S\_Box$ , yaitu  $S1\_Box$  untuk  $kelp\_1(s)$ , ..., dan  $S8\_Box$  untuk  $kelp\_8(s)$ , masing-masing kelompok ke  $S\_Box$  yaitu  $kelp\_1(s)$  ke  $S1\_Box$ ,  $kelp\_2(s)$  ke  $S2\_Box$ , ... .. $kelp\_8(s)$  ke  $S8\_Box$ . Substitusi: namakan bit kelompok dengan  $b1k2b3b4b5b6$  pada masing-masing kelompok, bilangan hexa dari  $b1b6$  menentukan baris dan bilangan hexa dari  $b2b3b4b5$  menentukan kolom pada tabel  $S\_Box$  yang bersesuaian dengan kelompoknya. Ubah bilangan hexa baris dan kolom tabel  $S\_Box$  ke biner 4 bit untuk masing-masing kelompok, diperoleh  $Bsi$  32 bit

Menentukan  $f(ri-1,ki)=P(Bsi)$ , yaitu permutasi  $P$  terhadap  $Bsi$ .

Analog: untuk ciphertext yang lainnya. Dari iterasi ke<sub>16</sub> proses dekripsi diperoleh  $(l16(i),r16(i))$  dengan  $r16(i)=l15(i)\oplus P(Bs16(i))$  dan  $l16(i)=r15(i)$ .

*Plaintext biner*=  $IP^{-1}(l16(i), r16(i))$ .

#### d. Proses IDWT

Dari hasil dekripsi yaitu  $(l16(i),r16(i))$  berbentuk biner,selanjutnya  $(l16(i),r16(i))$  diubah ke bentuk desimal, namakan  $xgryhsl$  untuk grey dan untuk  $(R,G,B)$  adalah  $xrhsl,xghsl$  dan  $xbhsl$ . Proses transformasi *IDWT* digunakan untuk proses pengembalian citra ke bentuk sebelum ditransformasi *DWT*.

```
Level_1: gray : [cA1]=xdwt;
x=idwt2(cA1,cH1,cV1,cD1, 'haar');
RGB, matriks m_r: [cA1]=xrdwt;
xr=idwt2(cA1,cH1,cV1,cD1, 'haar');
Level_2: gray:[cA2]=xdwt;
sx2=size(xdwt);
xhsl1=idwt2(xdwt,cH2,cV2,cD2,
'haar',sx2); sx1=2*size(xhsl1);
xhsl=idwt2(xhsl1,cH1,cV1,cD1,
'haar',sx1);
RGB, matriks m_r:
[cA2]=xrdwt; sx2=size(xrhsl2);
xrhsl1=idwt2(xrdwt,cH2,cV2,cD2,
'haar',sx2);
sx1=2*size(xrhsl1);
xrhsl=idwt2(xrhsl1,cH1,cV1,cD1,
'haar',sx1);
```

#### e. Proses Rekontruksi

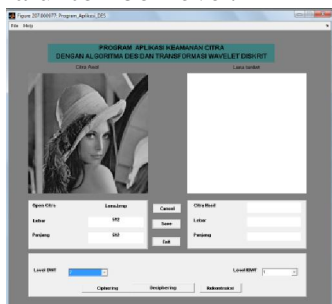
Proses rekontruksi, merupakan proses pengembalian citra hasil dari proses *IDWT*, ke bentuk citra awal. Untuk kasus citra *RGB*, perlu dilakukan penggabungan kembali matriks citra hasil dari proses *DWT*, enkripsi, dekripsi, dan *IDWT* dengan matriks-matriks lain yang tidak diambil didalam proses *DWT* Pada kasus matriks  $m\_r$  diambil untuk proses *DWT* dan misalkan hasilnya  $xrdwt$ , setelah melalui proses-proses enkripsi,dekripsi dan *IDWT* misalkan hasilnya  $xrhasil$ , maka  $xrhasil$  harus digabungkan dengan matriks  $m\_g$  dan  $m\_b$ .

#### 4. HASIL DAN PEMBAHASAN

##### 4.1 Hasil Penelitian

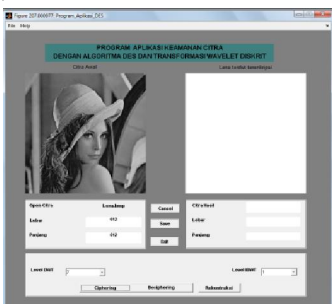
Hasil dari penelitian ini, disajikan dalam bentuk hasil tiap-tiap proses program aplikasi dan untuk hasil akhir yaitu program aplikasi keamanan citra adalah keseluruhan hasil proses disajikan dalam bentuk hasil antar muka . Berikut ini adalah salah satu contoh hasil tampilan untuk Lena.bmp .

a. Gambar 8, pada *axes1* merupakan gambar hasil tampilan *open* citra yaitu untuk citra awal *Lena.bmp* melalui perintah *open* citra pada menu *File* atau tombol *open* citra dan pada *axes2* merupakan gambar *Lena\_terdwt* yang merupakan hasil setelah melakukan perintah pemilihan *level\_7* transformasi *DWT* melalui tombol *level*.



Gambar 8 Axes 1 Citra Lena Awal, pada Axes2 *Lena\_terdwtlv17*

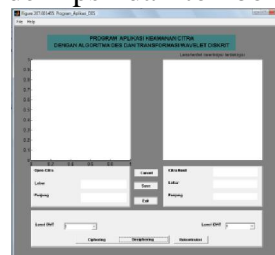
b. Gambar 9 pada *axes1* citra awal dan pada *axes2* merupakan gambar tampilan citra hasil *Lena\_terdwtlv17\_terenkripsi* melalui perintah enkripsi dari tombol Enkripsi.



Gambar 9 Axes 2 Lena awal dan *Lena\_terdwtlv17\_terenkripsi*

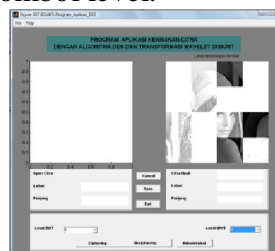
c. Gambar 10 pada *axes1*, gambar *Lena\_terdwtlv17\_terenkripsi* dan pada *axes2*, gambar *Lena\_terdwtlv17\_*

*terenkripsi\_terdekripsi*. hasil dari perintah dekripsi dari tombol dekripsi.



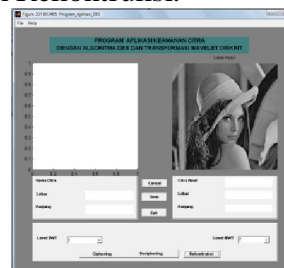
Gambar 10 *Axes1* Lena Awal dan *Lena\_terdwtlv17\_terenkripsi* ,pada *Axis 2* *Lena\_terdwtlv17\_terenkripsi\_terdekripsi*

d. Gambar 11 pada *axes1*, *Lena\_terdwtlv17\_* *terenkripsi\_terdekripsi* dan pada *axes2* gambar *Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_teridwt*. hasil dari perintah *IDWT* tombol *level*.



Gambar 11 *Axis 2* *Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_teridwt*

e. Gambar 12 pada *axes1*, gambar *Lena\_terdwtlv17\_terenkripsi* dan pada *axes2* merupakan gambar citra hasil *Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_teridwt\_terekonstruksi*, dari perintah Rekonstruksi.



Gambar 12 *Axis2* *Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_teridwt\_terekonstruksi*

##### 4.2 Pembahasan Hasil Penelitian

Pembahasan hasil penelitian ini, merupakan analisis hasil komponen-komponen penyusun program aplikasi keamanan citra sebagai berikut.

a. Pembahasan Hasil Transformasi *DWT*

Gambar 8 axes 2 Lena\_terdwt hasil transformasi *DWT* dengan filter *Haar* pada level 7 berukuran 4x4 piksel, dari citra awal *Lena.bmp* yang berukuran 512x512. Semakin tinggi level yang diambil, maka semakin kabur/terkesan kosong image citranya. Hal ini dikarenakan jumlah piksel hasil kompresi *DWT* semakin sedikit, jika ukuran piksel image citra awal adalah  $m \times n$  maka jumlah piksel pada level  $k$  akan berkurang menjadi  $1/2^k \times (m,n)$ .

#### b. Pembahasan Hasil Algoritma Enkripsi DES

Contoh: Lena\_terdwt level 7, menghasilkan image berukuran 4x4 piksel. Panjang bit plaintext binernya adalah 256 bit, banyak kelompok  $256/64$  ada 4 kelompok yang masing-masing panjangnya 64 bit, setelah melalui 16x iterasi:

*ciphxt* =

65386	23234	32854	10303
27913	1714	45215	7191
10506	50795	56209	57549
53699	30124	12749	59252

*ciphxt*=[bars1,bars2,bars3,bars4;bars5, bars6,bars7,bars8;bars9,bars10,bars11,bars12; bars13,bars14,bars15,bars16];

*gambciph*=double(*ciphxt*);*gbr\_enk*=uint8(*gambciph*); *imshow*(*gbr\_enk*);

gambar 4.2 axes 2

Lena\_terdwt\_terenkripsi yang berukuran 4x4 yaitu sama dengan ukuran gambar *lena\_terdwtlv17*.

#### c. Pembahasan Hasil Algoritma Dekripsi DES

Dari contoh: Lena\_terdwt level 7 enkripsi Setelah melalui 16xiterasi menghasilkan *plaintext*:

*pltxtxsl* =

61649	31096	9000	62277
40021	28903	49946	55050
29539	43941	25015	3635
30182	42088	11066	39337

*pltxtxsl*=[bars1,bars2,bars3,bars4;bars5, bars6,bars7,bars8;bars9,bars10,bars11,bars12; brs13,bars14,bars15,bars16]';

*gamb\_dek*=double(*pltxtxsl*);

*gbr\_dek*=uint8(*gamb\_dek*);

*imshow*(*gbr\_dek*); Gambar 9 axes 2

Lena\_terdwtlv17\_terenkripsi\_terdekripsi,

berukuran 4x4, sama dengan ukuran Lena\_terdwtlv17\_terenkripsi

#### d. Pembahasan Hasil Transformasi IDWT

Dari Lena\_terdwtlv17\_terenkripsi\_terdekripsi, kemudian dilakukan pengembalian piksel acak ke sebelum terenkripsi dan terdekripsi, dengan *IDWT*:

```
pl=imread('lena.bmp');
xdwt=[cA7]; [cA7]=pltxtxsl;
sx7=2*size(pltxtxsl);
xhsl6=idwt2(pltxtxsl,cH7,cV7,cD7,'haar',
sx7);
```

```
sx6=2*size(xhsl6);
```

```
xhsl5=idwt2(xhsl6,cH6,cV6,cD6,'haar',
sx6);
```

```
sx5=2*size(xhsl5);
```

```
xhsl4=idwt2(xhsl5,cH5,cV5,cD5,'haar',
sx5);
```

```
sx4=2*size(xhsl4);
```

```
xhsl3=idwt2(xhsl4,cH4,cV4,cD4,'haar',
sx4);
```

```
sx3=2*size(xhsl3);
```

```
xhsl2=idwt2(xhsl3,cH3,cV3,cD3,'haar',
sx3);
```

```
sx2=2*size(xhsl2);
```

```
xhsl1=idwt2(xhsl2,cH2,cV2,cD2,'haar',
sx2);
```

```
sx1=2*size(xhsl1);
```

```
xhsl=idwt2(xhsl1,cH1,cV1,cD1,'haar',sx1;
```

```
gbre_hsl=double(xhsl);
```

```
gamb_hsl=uint8(gbre_hsl);
```

```
imshow(gamb_hsl);
```

Gambar 10 axes 2, Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_teridwt berukuran 512x512 piksel, merupakan hasil transformasi *IDWT* dari Lena\_terdwtlv17\_terenkripsi\_terdekripsi.

#### e. Pembahasan Hasil Rekontruksi

Rekontruksi, merupakan proses akhir, yaitu proses mengembalikan citra\_terdwt\_terenkripsi\_terdekripsi\_teridwt ke citra awal, yaitu dari

Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_teridwt, melalui perintah:

```
xhsl=pl;gbre_hsl=double(xhsl);gamb_hsl=
uint8(gbre_hsl); imshow(gamb_hsl);
```

Gambar 4.4 axes 2, hasil rekontruksi Lena\_terdwtlv17\_terenkripsi\_terdekripsi\_



teridwt, berukuran 512x512 sama dengan ukuran citra awal.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan:

- a. Kompresi dengan menggunakan transformasi *DWT* dengan filter Haar level yang semakin tinggi, menghasilkan image citra semakin kabur/terkesan kosong, proses enkripsi dan dekripsi menjadi sederhana.
- b. Enkripsi *DES* setelah melalui kompresi, hanya melakukan pengacakan posisi piksel-piksel. Begitu pula dekripsi *DES*, hanya melakukan pengacakan kembali posisi piksel-piksel terenkripsi ke bentuk sebelum dienkripsi yaitu ke posisi citra\_terdwt. Hal ini menunjukkan bahwa program aplikasi ini aman digunakan untuk menyimpan citra maupun mengirim citra lewat email.
- c. Nirkompresi melalui transformasi *IDWT* dengan filter *Haar*, mengembalikan posisi piksel-piksel ke citra awal sebelum melakukan transformasi *DWT*. Untuk mengisi posisi piksel-piksel kosong hasil *IDWT*, dilakukan rekonstruksi piksel-piksel hasil dengan piksel-piksel pendukung citra awal. Hasilnya sama dengan citra awal, hal ini menunjukkan bahwa program aplikasi ini tidak merubah citra aslinya.
- d. Citra terenkripsi tersimpan dengan aman, jika dikirim lewat email akan terjamin kerahasiaanya dan dari akibat kompresi maka ruang simpan atau ruang email kirim membutuhkan ruang yang jauh lebih kecil dibanding dengan ruang simpan atau ruang email kirim citra aslinya.

### 5.2 Saran

Bagi pengguna atau peneliti lain, disarankan :

- a. Sebaiknya untuk pemilihan algoritma enkripsi, pilih algoritma enkripsi yang tidak membutuhkan iterasi yang banyak, tentu dengan keamanan yang baik.
- b. Perlu melakukan penelitian yang berkaitan dengan keberadaan posisi piksel-piksel nirkompresi setelah melakukan kompresi yang menggunakan transformasi *DWT*.

## 6. DAFTAR PUSTAKA

- [1]. A. Dony, (2006), *Kriptografi Keamanan Data Dan Komunikasi*, Graha Ilmu. Yogyakarta.
- [2]. Krisnawati, (2006), *Transformasi Fourier Dan Transformasi Wavelet Pada Citra*. Dasi. Vol 7. No4.
- [3]. Munir, R. (2004), *Pengolahan Citra Digital*, Informatika Bandung.
- [4]. Munir, R. (2006), *Kriptografi*. Informatika Bandung.
- [5]. R. Budi, (1998-2005), *Keamanan Sistem Informasi Berbasis Internet*, PT Insan Infonesia-Bandung & PTINDOCISC-Jakarta.
- [6]. S.F., Isma. (2008), *Implementasi Steganografi Pada Video jenis AVI Menggunakan Transformasi Wavelet Diskrit*, IT Telkom Buah Batu. Bandung.
- [7]. S., Saleh. (2006), *Penerapan Algoritma Des Dalam Sistem Keamanan Data dengan penambahan Password Terjadwal*.
- [8]. S., William, (2002), *Komunikasi Data dan Komputer: Jaringan Komputer*, Thamir Abdul Hafedh Al\_Hamdany (Penterjemah). Salemba Teknik. Jakarta.