SEMINAR NASIONAL INOVASI DAN APLIKASI
TEKNOLOGI DI INDUSTRI (SENIATI) 2016
ISSN : 2085-4218

Merancang Citra Watermark Menggunakan DCT dan Terenkripsi – Kompresi Teks Menggunakan RSA-HUFFMAN Code

Chandra Buana Putrana[1,*], Gelar Budiman[1], I Nyoman Apraz Ramatryana[1]
1 Telkom University, Fakultas Teknik Elektro, Terusan Buah Batu, Bandung
* E-mail : canzdoankz@gmail.com

Abstrak. Seiring dengan perkembangan internet yang semakin cepat, penggunaan watermarkpada produk digital sudah menjadi hal yang popular dipakai. Kegunaannya sudah banyak diaplikasikan seperti pemberian hak cipta atau bukti kepemilikan, memonitar *Broadcast,* pengecekan transaksi, dan juga pengidentifikasian kontan. Masalah yang sekarang timbul adalah banyaknya media sosial yang membantu mempercepat peredaran produk digital, tetapi tidak ada pembuktian hak cipta, sehingga dilain pihak dapat merugikan seniman digital di era ini karena tidak adanya pembuktian hak cipta tersebut. Penulis akan merancang suatu sistem pemberian watermark pada gambar menggunakan metode *DCT* yang dipadukan dengan enkripsi – kompresi teks menggunakan *RSA-Huffman Code* yang dirancang pada perangkat telepon seluler dengan sistem operasi Android. Akhir yang diharapkan dapat dihasilkan suatu watermark yang tahan terhadap serangan seperti *noise*, kompresi gambar, dan memiliki PSNR watermark gambar adalah sebesar 30%.

Kata Kunci: Watermark, Enkripsi, Kompresi, DCT, RSA , Huffman Code, Android.

1. Pendahuluan

Dewasa ini, *digital watermarking* sudah menjadi popular sebab karena meningkatnya pengiriman data yang dikirim secara digital, mulai dari maraknya social media seperti twitter, facebook, instagram, path, dan lainnya sehingga mudahnya karya seseorang diambil dengan mudah. Dengan adanya teknik ini, maka dapat membantu untuk menjaga hak milik seseorang sehingga tidak mudah disalah gunakan [1].

Pada penelitian di [2] Steganografi menggunakan metode LSB dengan enkripsi-kompresi menggunakan TEA-LZW. Di [2] dijelaskan bahwa dengan metode LSB ini rentan terhadap gangguan. Maka dari itu saya pada tugas akhir ini mencoba menggunakan metode DCT yang dipadukan dengan enkripsi-kompresi menggunakan RSA-Huffman Code yang diharapkan akan memberikan performansi yang lebih dari dibandingkan LSB.

.

2. Dasar Teori

2.1. Digital Watermarking *[1]*

Digital Watermarking adalah teknik dimana data digital disisipkan kedalam *host data*dalam berbagai cara yang membuat data tersebut tidak terganggu oleh pemakaian *host data* , tidak dapat dihilangkan melalui proses biasa, trasmisi, dan atau rekaman dari *host data*, dan dapat dibaca ulang dengan *watermark detectors* yang tepat.

2.2. Discrete Cosine Transformation (DCT)

Transformasi pada citra sangat penting dalam pengolahan sinyal digital, dengan ini dapat memperoleh hasil yang lebih. Contohnya Transformasi *Fourier* digunakan secara efektsif untuk menghitung konvolusi dari citra, atau *Discrete Cosine Transform* dapat digunakan untuk mengurangi ruang secara drastis dari gambar tanpa diketahui kualitas yang hilang [3]

ISSN : 2085-4218

Untuk 2-D DCT rumus demikian [4] :

$$c(u,v) = a(u)a(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (1)$$

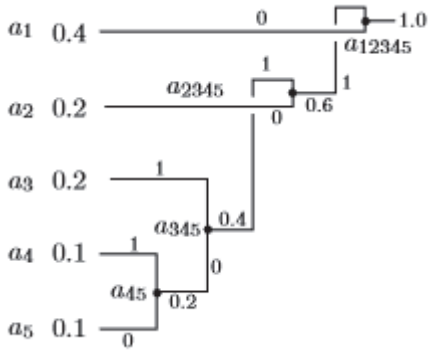
$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & untuk\ u = 0 \\ \sqrt{\frac{2}{N}} & untuk\ u \neq 0 \end{cases} \quad (2)$$

Untuk 2-D IDCT

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} a(u)a(v)\,c(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (3)$$

2.3. Huffman Code *[5]*

Kode *Huffman* adalah suatu algoritma dimana mengurutkan probabilitas kode-kode yang dihasilkan dan menyusun dengan cara menambahkan dengan probabilitas simbol diatasnya kemudian menghilangkannya dari list sehinnga membentuk ukuran seperti pohon. Ketika listnya berkurang hingga hanya ada 1 simbol, pohon sudah selesai. Pohon ini yang akan menjadi kode dari masing-masing simbol tersebut dengan mengurutkannya.



Gambar 2 Contoh *Huffman Code*

RSA [5]

Algoritma RSA dibuat oleh tiga orang peneliti dari MIT (Massachussets Institute of Technology) pada tahun 1976, yaitu Ron Rivest, Adi Shamir dan Leonard Adleman yang menggantikan algoritma National Bureau of Standards (NBS). Algoritma ini menggunakan algoritma kunci publik, dimana antara enkripsi dan dekripsi menggunakan kunci yang berbeda. Untuk rumus enkripsi :

$$C = M^e mod(n) \quad (4)$$

Rumus dekripsi :

$$M = C^d mod(n) \quad (5)$$

Dimana C merupakan *chiper text,* M adalah data inputan, e dan d merupakan bilangan bulat dimana :

$$e.d = 1 \mod(\varphi(n)) \quad (6)$$

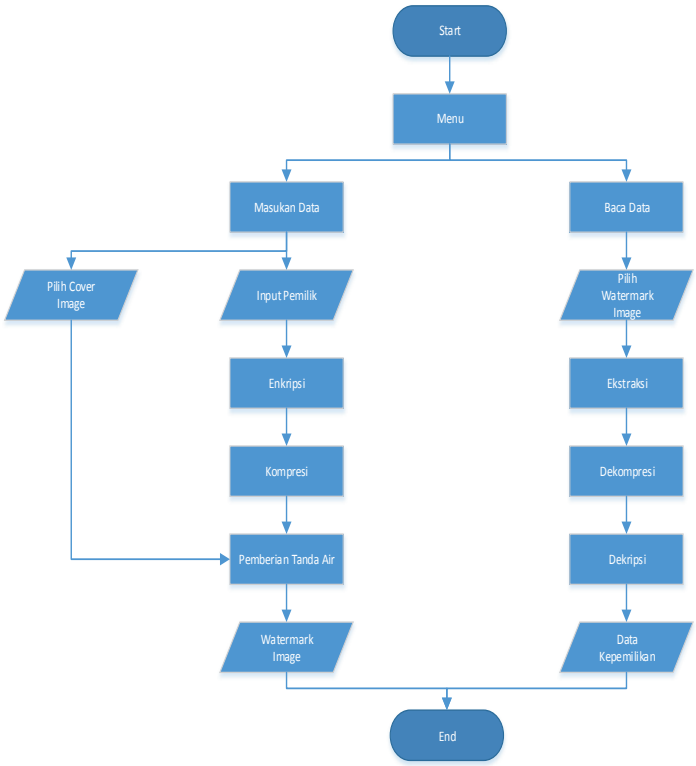
$$\varphi(n) = (p-1).(q-1) \quad (7)$$

p dan q merupakan bilangan prima

ISSN : 2085-4218

3. Pemodelan Sistem

3.1 Proses Embed dan Ekstraksi



Gambar 1 Diagram AlirSistem

Alur kerja dari aplikasi watermark dengan teks yang terenkripsi dan kompresi ini dapat dijabarkan dalam penjelasan berikut :

1. Pada Bagian masukan data (embeding) pertama data yang ingin dimasukan kedapat *cover image* terlebih dahulu dienkripsi menggunakan RSA dan kemudian dikompresi dengan table *Huffman* kemudian dapat diembed atau disisipkan kedalam cover image menggunakan metode DCT pada tertentu dengan kondisi 1 maka memberi nilai positif dan 0 memberi nilai negatif
2. Pada bagian baca data atau ektraksi pertama data yang sudah di embed kita baca dengan cara merubah domain gambar menggunakan DCT dan melihat nilai dari pixel yang ditentukan, kemudian setelah dapat nilainya makanya dilakukannya dekompresi lalu kemudian dekripsi hingga mendapatkan informasi watermark yang telah disisipkan

4. Pengujian dan Analisis

4.1 Parameter Pengujian

Parameter-parameter yang akan diukur dalam pengujian sistem adalah sebagai berikut :

1. Penilaian hasil steganografi secara subjektif
   Kriteria kebenaran subjektif dilakukan dengan menilai citra hasil steganografi dengan menanyakan secara langsung (*polling)* kepada orang-orang tentang kualitas hasil *stego image*. Metode ini biasanya lebih tepat atau lebih cocok.Penilaian dapat dilakukan dengan membandingkan *watermark image* dengan *cover image*, kemudian dibuat suatu skala penilaian dimana setiap skala berkaitan dengan kualitas. Pada tugas akhir ini *polling* dilakukan terhadap 30 responden dengan skala penilaian 5. Hasil akhir penilaian ini adalah MOS (*Mean Opinian Score*).
2. Penilaian hasil steganografi secara objektif
   Kriteria Penilaian secara objektif berdasarkan pada proses perhitungan secara metematis, dengan menggunakan parameter *Mean Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR). *Mean*

ISSN : 2085-4218

Square Error (MSE) adalah rata-rata nilai *error* antara citra *cover* dengan citra *stego*. Secara matematis, *Mean Square Error* (MSE) dapat dirumuskan sebagai berikut :

$$MSE = \frac{1}{mxn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j) - I^{'}(i,j)\right]^2 \quad (4.4)$$

Keterangan :
- I(i,j) = citra *cover*
- I'(I,j) = citra *stego*
- m = lebar citra
- n = tinggi citra

Peak Signal to Noise Ratio (PSNR) adalah perbandingan antara harga maksimum dari *pixel* citra yang terbentuk dengan *Mean Square Error* (MSE). Secara matematis, *Peak Signal to Noise Ratio* (PSNR) dapat dirumuskan sebagai berikut :

$$PSNR = 10x\log_{10}\left(\frac{255^2}{MSE}\right) \quad (4.5)$$

Keterangan :
- MSE = *Mean Square Error*
- PSNR dinyatakan dalam satuan desibel(dB)

Berikut adalah rentang PSNR dan deskripsi yang digunakan sebagai acuan pada Tugas Akhir ini.

Tabel 1. Nilai PSNR [www.cctv-information.co.uk/constant2/sn_ratio.html]

PSNR (dB)	*Picture Quality*
60	*Excellent, no noise apparent*
50	*Good, a small amount of noise but picture quality good*
40	*Reasonable, fine grain or snow in the picture, some fine detail lost*
30	*Poor picture with a great deal of noise*
20	*Unusable*

3. Kinerja terhadap serangan
   Pada penilaian kinerja terhadap serangan, ada parameter yang dicobakan terhadap gambar tersebut yaitu pengaruh dari amplituda terhadap ketahanan data, baik secara bit maupun karakter yang terdeteksi nanti. Penilaian ini dilakukan dengan cara membandingan banyaknya data yang diterima dengan data asli yang disisipkan. Pemilihan serangan menggunakan *noise noise* ini karena serangan tersebut memiliki kecenderungan distribusi yang sudah hampir mewaliki serangan yang ada.

4.2 Pengujian sistem

Untuk mengukur penilaian hasil watermark secara subjektif menggunakan MOS (*Mean Oponian Score*). MOS pada pengujian ini didapat dari hasil kuisioner kepada 30 responden. *Watermark image* yang dinilai adalah sebanyak 30 dengan amplituda data yang disisipkan berbeda-beda dan 5 gambar dengan resolusi citra yang digunakan sama. Karena sistem yang dibuat berjalan pada perangkat Android maka penilaian hasil *Watermark image* dilakukan pada perangkat Android juga. Hasil MOS dapat dilihat pada tabel berikut :

Tabel 2. Perbandingan nilai MOS

File	Amplituda			
	15	30	45	60
data_1.png	4,633333	4,6	4,5	4,233333
data_2.png	4,633333	4,7	4,333333	3,9
data_3.png	4,6	4,433333	4,066667	3,9
data_4.png	4,7	4,6	4,033333	3,666667
data_5.png	4,866667	4,333333	3,833333	3,633333

Nilai PSNR yang dihitung berdasarkan gambar yang diberi watermark, dengan *cover image.* Gambar pengujian diambil 5 sampel gambar dengan jumlah bit yang disisipkan sama setiap gambar. Parameter pengujian dalam pengujian ini adalah pengaruh dari amplituda data terhadap nilai PSNR yang dihitung. Berdasarkan perhitungan, didapat data sebagai berikut:

ISSN : 2085-4218

Tabel 3 Perbandingan nilai PSNR gambar berdasarkan amplituda

File asli	File sisip	Amplituda	MSE	PSNR(dB)
data_1.png	latih_char1_8_15.png	15	35,55058	32,62234
data_1.png	latih_char1_8_30.png	30	112,0193	27,63788
data_1.png	latih_char1_8_45.png	45	236,0744	24,40031
data_1.png	latih_char1_8_60.png	60	405,347	22,05253
data_2.png	latih_char2_8_15.png	15	38,48114	32,27832
data_2.png	latih_char2_8_30.png	30	119,6083	27,35319
data_2.png	latih_char2_8_45.png	45	253,6247	24,08889
data_2.png	latih_char2_8_60.png	60	438,0911	21,71516
data_3.png	latih_char3_8_15.png	15	40,78532	32,02576
data_3.png	latih_char3_8_30.png	30	119,6232	27,35265
data_3.png	latih_char3_8_45.png	45	248,7363	24,17341
data_3.png	latih_char3_8_60.png	60	421,4618	21,88322
data_4.png	latih_char4_8_15.png	15	32,23879	33,04702
data_4.png	latih_char4_8_30.png	30	108,8394	27,76294
data_4.png	latih_char4_8_45.png	45	229,5955	24,52117
data_4.png	latih_char4_8_60.png	60	388,7249	22,23438
data_5.png	latih_char5_8_15.png	15	36,81004	32,47114
data_5.png	latih_char5_8_30.png	30	107,6253	27,81166
data_5.png	latih_char5_8_45.png	45	222,0392	24,666
data_5.png	latih_char5_8_60.png	60	377,8081	22,35809

Pada pengujian pemberian *Gaussian Noise* menggunakan beberapa parameter pengujian. Parameter tersebut adalah amplituda data, *vairance* dengan nilai sebesar 0.001, 0.002, 0.003, 0.004, 0.005 dan juga *mean* = 0. Setelah diberikan *noise* tersebut lalu gambar masuk ke sistem untuk diekstrak apakah data dapat kembali terbaca atau tidak. Pada hasil pengujian tersebut, didapat data sebagai berikut:

Tabel 4 Tabel Hasil pembacaan data *Noise Gaussian*

	Variance				
Nama File	0.001	0.002	0.003	0.004	0.005
latih_char1_8_15.png	0,971662	0,897003	0,846866	0,837057	0,782016
latih_char1_8_30.png	1	0,996185	0,983106	0,967847	0,951499
latih_char1_8_45.png	1	1	0,99891	0,99564	0,99346
latih_char1_8_60.png	1	1	1	1	0,99891
latih_char2_8_15.png	0,970572	0,907902	0,859401	0,813079	0,789646
latih_char2_8_30.png	0,999455	0,99782	0,984741	0,972752	0,948774
latih_char2_8_45.png	1	1	1	0,994005	0,990736
latih_char2_8_60.png	1	1	1	1	0,99891
latih_char3_8_15.png	0,968392	0,901362	0,852316	0,825613	0,796185
latih_char3_8_30.png	1	0,998365	0,984196	0,963488	0,944959
latih_char3_8_45.png	1	1	0,99891	0,99346	0,994005
latih_char3_8_60.png	1	1	0,999455	0,999455	0,999455
latih_char4_8_15.png	0,961308	0,890463	0,843597	0,80327	0,779292
latih_char4_8_30.png	0,99891	0,986921	0,959673	0,947684	0,92861
latih_char4_8_45.png	1	1	0,99455	0,984741	0,977112
latih_char4_8_60.png	1	1	0,999455	0,99782	0,989101
latih_char5_8_15.png	0,9297	0,876294	0,817439	0,767302	0,758583
latih_char5_8_30.png	0,995095	0,973842	0,966757	0,930245	0,911717
latih_char5_8_45.png	0,999455	0,99673	0,991281	0,981471	0,966213
latih_char5_8_60.png	1	0,999455	0,996185	0,990736	0,985831

Pada pengujian serangan *Salt and Pepper Noise* dilakukan dengan beberapa parameter. Parameternya yaitu *noise density*, dan amplituda data. *Noise density* diberikan sebanyak 5 yaitu 0.001, 0.002, 0.003, 0.004, 0.005 dan amplituda 15, 30, 45, 60 serta pengujian menggunakan 5 sample gambar dengan jumlah bit sisipan sama. Dari pengujian tersebut didapatkan data sebagai berikut:

ISSN : 2085-4218

Tabel 5 Tabel Hasil pembacaan data *Noise Salt and Pepper*

	Variance				
Nama File	0.001	0.002	0.003	0.004	0.005
latih_char1_8_15.png	0,99455	0,971662	0,958038	0,954768	0,93733
latih_char1_8_30.png	0,996185	0,995095	0,983651	0,986921	0,975477
latih_char1_8_45.png	0,999455	0,99891	0,998365	0,99673	0,994005
latih_char1_8_60.png	1	1	1	0,99891	0,99891
latih_char2_8_15.png	0,985286	0,974932	0,969482	0,946594	0,946049
latih_char2_8_30.png	0,99673	0,992916	0,985831	0,988011	0,979837
latih_char2_8_45.png	0,999455	0,998365	0,99455	0,997275	0,99673
latih_char2_8_60.png	1	1	1	1	1
latih_char3_8_15.png	0,99346	0,977112	0,954223	0,963488	0,949864
latih_char3_8_30.png	0,99673	0,990191	0,988011	0,974932	0,974932
latih_char3_8_45.png	0,999455	0,99782	0,99564	0,990191	0,986376
latih_char3_8_60.png	0,999455	0,999455	0,99891	0,999455	1
latih_char4_8_15.png	0,986376	0,992916	0,969482	0,941144	0,946594
latih_char4_8_30.png	0,989646	0,981471	0,982561	0,971117	0,972752
latih_char4_8_45.png	0,99891	0,99564	0,986921	0,985286	0,973297
latih_char4_8_60.png	0,997275	0,994005	0,995095	0,992371	0,99455
latih_char5_8_15.png	0,99673	0,973297	0,963488	0,950954	0,933515
latih_char5_8_30.png	0,99346	0,987466	0,988556	0,974932	0,969482
latih_char5_8_45.png	0,995095	0,988011	0,983651	0,984741	0,970572
latih_char5_8_60.png	0,995095	0,995095	0,992371	0,988011	0,988011

5. Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan pada sistem watermark menggunakan metode DCT dengan teks terenkripsi dan kompres RSA-*Huffman Code* ini, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Pada sistem ini , nilai MOS yang diperoleh tiap amplituda menggambarkan bahwa semakin besar nilai amplituda maka nilai MOS akan semakin kecil, maka dari itu pemilihan nilai amplituda yang diambil untuk pengimplementasian adalah 30 karena pada nilai tersebut nilai MOS tiap gambar berada rata-rata diatas nilai 4.
2. Untuk nilai PSNR dapat dilihat bahwa nilai yang didapat didapat kurang dari 40dB. Dari nilai tersebut menandakan bahwa dalam sistem watermark ini mengakibatkan *noise* yang cukup banyak terhadap *cover image* karena data disisipkan kesmua bagian gambar sehingga pixel nilai pixel dalam gambar berubah sebagian besar.
3. Pada pengujian dengan serangan *noise gaussian* data bertahan baik pada amplituda 60 sampai *variance* 0.005 dan semakin kecil amplituda, semakin kurangnya ketahanan dari data tersebut.
4. Dalam pengujian dengan serangan *noise salt and pepper* data akan semakin bertahan dengan baik jika amplituda semakin tinggi. Dapat dilihat bahwa semakin tinggi amplituda nilai dari 1-BER akan semakin besar, yang menandakan bahwa semakin banyak bit yang berhasil dideteksi.

6. Daftar Referensi

[1] Dongyang Teng, Renghui Shi, XIaoqun Zhao "DCT Image Watermarking Technique Based on the Mix of Time-domain",Tongji University, Shanghai,China.
[2] Made Sumarsana Adi Putra "Implementation of Steganography using LSB with Encrypted and Compressed Text using TEA-LZW on Android", Telkom University,Indonesia.
[3] Deepak Singla,Rupali Syal "Data Security Usin LSB & DCT Steganography In Images",PEC University of Technology,India.
[4] Ali Khayam ,S. (2003), Michigan State University, The Discrete Cosine Wavelets.
[5] A Concise Introduction to Data Compression Salomon,D. 2008 XIV, 314 p. 89 illus. ISBN: 978-1-84800-071-1
[6] Ronald L. Rivest, MIT Laboratory for Computer Science and Burt Kaliski, RSA Laboratories. 10-12-2003
[7] Safaat, Nazruddin.2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung : Informatika