Volume 2 No. 2, Desember 2017
http://ejournal.ust.ac.id/index.php/Jurnal_Means/
MEANS (Media Informasi Analisa dan Sistem)
p-ISSN : 2548-6985, e-ISSN : 2599-3089

Aplikasi Otentikasi Citra Digital Dengan Metode Adaptive Data Hiding

Damarlin Zalukhu
STMIK Budi Darma Medan, Jl. Sisimangaraja No.338 Simpang Limun Medan
http : //www.stmik-budidarma.ac.id // Email : Damarzal@yahoo.com

ABSTRAK

Problems that often occur in internet services make data security not guaranteed. To provide content authentication and duplicate protection from digital data, two complementary techniques have been developed, namely encryption and watermarks. Encryption techniques are used to secure digital data during transmission from sender to receiver. Because digital images can be easily reproduced, cryptosystems cannot solve this problem perfectly. The second security measure that completes encryption is to attach a secret signal, a watermark, directly to the original data so that it remains available during use. Watermarking cannot prevent duplication, modification and distribution of digital media by itself, so it must be combined with cryptography. Embedding is done by modifying the average pixel intensity value of each block in the range determined by the contrast value for a particular block. This reduces the effect of modification as felt by the human eye. The extraction procedure calculates and compares the number of pixel values for blocks of the original image and watermark. The results show the robustness of the scheme towards general image processing operations such as cropping, modification, low pass filters, median pass filters, minimized and lossy JPEG compression with various quality index factors. The results also illustrate that the watermark is safe, restored and recognized even after the watermark image has been damaged, forged and modified by a general image processing operation. A comparative study of the scheme proposed with the existing scheme has also been carried out to observe the strength of the scheme.

Kata kunci : Adaptive Data Hiding (ADH), Joint Photographic Expert Group (JPEG), Message Digest (MD), satu arah Hash, Spasial Domain.

PENDAHULUAN

Banyak *vendor* dan pengembang komersial menggunakan internet untuk mengirimkan data dan melakukan transaksi bisnis. Selama transmisi dari data digital, layanan seperti permintaan video, pertukaran data elektronik dan *online shopping* menghadapi dua permasalahan. Permasalahan pertama adalah bahwa layanan ini rentan terhadap akses yang mudah dari pengguna ilegal sehingga sekuritas data menjadi tidak aman. Permasalahan lainnya adalah layanan ini mudah untuk diduplikasi dan didistribusikan ulang. Untuk menyediakan otentikasi isi dan proteksi duplikasi dari data digital, dua teknik yang saling melengkapi telah dikembangkan, yaitu enkripsi dan *watermark* [7]. Teknik enkripsi digunakan untuk mengamankan data digital selama transmisi dari pengirim ke penerima.

Skema *Adaptive Data Hiding* (ADH) ini akan menyembunyikan sebuah citra *watermark* hitam putih dalam citra asli sehingga menghasilkan sebuah citra *watermark* baru yang secara kasat mata kelihatan sama dengan citra asli [4]. Data yang akan ditempelkan pada citra asli tergantung pada kunci yang dimasukkan, karena citra *watermark* yang akan ditempelkan pada citra asli akan diproses dengan menggunakan nilai *hash* dari kunci *input*. *Joint Photographic Experts Group* (JPEG) adalah standar citra digital yang digunakan secara luas pada *World Wide Web* karena rasio citra digitalnya bagus dan kualitas juga.

Skema ini disebut adaptif karena *watermark* yang dihasilkan adalah aman, dapat diperoleh dan dikenali kembali, sekalipun citra *watermark* tersebut telah dimodifikasi dengan menggunakan operasi pengolahan citra yang umum digunakan. Skema ini menitikberatkan pada proteksi *copyright*, otentikasi isi, sekuritas dan *robustness*. Otentikasi isi dan proteksi *copyright* dengan sifat *robustness* dari citra digital tidak bergerak dapat dicapai dengan menggunakan sifat *watermarking* pada domain spasial sedangkan sekuritas dari skema dapat dicapai dengan menggunakan fungsi kriptografi hash satu arah.

Berdasarkan uraian latar belakang di atas, maka yang menjadi permasalahan adalah:

1. Bagaiman teknik dalam mengotentikasikan citra digital?
2. Bagaimana menerapkan metode *Adaptive Data Hiding* untuk mengotentikasikan suatu citra digital?
3. Bagaimana merancang aplikasi otentikasi citra digital?

Masalah yang akan dibahas dalam penelitian ini mencakup:

1. Input program adalah citra dalam format BMP dan JPEG.
2. *Format citra* hasil output sesuai dengan format citra input.
3. Ukuran citra asli yang dapat diproses dengan batasan minimal 100 x 100 *piks*el dan maksimal 1000 x 1000 *piksel*.
4. Citra asli berupa citra *grayscale* dan citra *watermark* berupa citra hitam putih. Citra asli berupa citra berwarna akan dikonversi menjadi citra *grayscale*.
5. Data kunci *input* bertipe data *string* dengan panjang maksimum 8 karakter.
6. Fungsi *hash* yang digunakan adalah fungsi SHA-1.

LANDASAN TEORI

2.1 Otentikasi

Otentikasi (*authentication*) merupakan sebuah metode untuk menyediakan jaminan bahwa informasi tidak dimanipulasi oleh pihak yang tidak mempunyai wewenang [1]. Otentikasi bersifat spesifik dalam topik keamanan yang berusaha dicapai. Contohnya meliputi pengendalian akses, otentikasi *entity*, otentikasi pesan, integritas data, *non-repudiation*, dan otentikasi kunci.

2.2 Fungsi Hash

Sebuah nilai *hash* dihasilkan oleh fungsi H dengan bentuk :

$$h = H(M) \ldots (1)$$

dimana M adalah pesan dengan panjang bebas, dan H(M) atau h adalah nilai *hash* dengan panjang tetap [1]. Nilai *hash* ini akan ditambahkan di awal pesan dan kemudian dikirimkan bersamaan. Penerima dari pesan itu akan melakukan otentikasi terhadap pesan itu dengan mengkomputasi nilai *hash* dan kemudian membandingkannya dengan nilai *hash* yang ada di awal pesan.

Agar fungsi *hash* bisa bertindak sebagai otentikasi pesan, fungsi *hash* H harus memenuhi syarat – syarat di bawah ini (Stalling, 1995) :

1. H dapat diterapkan pada blok data dengan ukuran berapa pun.
2. H menghasilkan keluaran dengan panjang yang tetap.
3. Relatif mudah untuk mengkomputasi H(x) untuk sembarang x.
4. Dengan mengetahui nilai *hash* y, tidaklah bisa secara komputasi untuk menemukan x yang memenuhi H(x) = y.
5. Dengan mengetahui x, tidaklah bisa secara komputasi untuk menemukan y ≠ x dengan H(y) = H(x).
6. Tidak bisa secara komputasi untuk menemukan pasangan (x,y) yang memenuhi H(x) = H(y).

Tiga syarat pertama berfungsi sebagai syarat penerapan praktis dari fungsi hash untuk otentikasi pesan.

Syarat keempat adalah syarat “**satu-arah**” (*one-way*). Mudah untuk menghasilkan kode nilai *hash* dengan diberikan sebuah pesan tetapi tidak bisa untuk menghasilkan pesan dengan diberikan kode nilai *hash*.

Syarat kelima menjamin bahwa dengan diberikan pesan asli tidak akan bisa ditemukan suatu pesan lain dengan nilai *hash* yang sama dengan nilai *hash* pesan aslinya. Syarat ini adalah untuk mencegah penipuan jika digunakan kode *hash* yang terenkripsi. Untuk kasus ini, pihak lawan bisa membaca pesan dan juga menghasilkan kode *hash*-nya. Tetapi karena pihak lawan tersebut tidak memiliki kunci untuk mendekripsikan kode *hash* tersebut, maka ia tidak akan bisa mengubah pesan tanpa diketahui. Jika syarat ini tidak dipenuhi maka pihak lawan bisa melakukan hal – hal berikut.

1. Mengintersepsi suatu pesan dengan kode *hash* terenkripsinya.
2. Menghasilkan kode *hash* yang tidak terenkripsi dari pesan tersebut.
3. Membuat pesan baru yang lain dengan kode nilai *hash* yang sama dan kemudian mengirimkannya kembali.

Jika kelima syarat di atas dipenuhi, maka fungsi *hash* tersebut dinamakan fungsi *hash* yang lemah. Jika syarat keenam juga dipenuhi, maka fungsi *hash* itu dianggap fungsi *hash* yang kuat.

Selain untuk otentikasi pesan, fungsi *hash* juga dapat diterapkan untuk menghasilkan kunci berbasis *passphrase*. Cara kerjanya adalah sebagai berikut :

1. Dengan menggunakan fungsi *hash* dihasilkan nilai *hash* dari *passphrase*.
2. Nilai *hash* dari *passphrase* dijadikan sebagai kunci untuk melakukan enkripsi / dekripsi.

Nilai fungsi *hash* merepresentasikan pesan yang lebih pendek dari dokumen dari mana nilai tersebut dihitung, nilai ini sering disebut *message digest*. *Message digest* dapat dianggap sebagai satu ”*digital fingerprint*” dari dokumen yang lebih panjang. Contoh fungsi *hash* yang terkenal adalah MD2, MD5, SHA dan HAVAL.

Peranan fungsi *hash* dalam kriptografi adalah dalam hal pengecekan kondisi terhadap integritas pesan dan tanda tangan digital. Suatu *digest* dapat dibuat publik tanpa menunjukkan isi dokumen dari mana *digest* tersebut diturunkan. Hal ini sangat penting dalam *digital timestamping* dimana dengan menggunakan fungsi *hash*, seseorang dapat memperoleh dokumen dengan stempel waktu (*document timestamped*) tanpa menunjukkan isi dari dokumen kepada penyedia layanan *timestamping*.

2.3 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek [2]. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan.

Pada saat ini citra sudah banyak yang diolah secara digital, karena dengan pengolahan citra secara digital citra akan mudah dimanipulasi, misalnya merubah warna, merubah ukuran merubah kecerahan warna ataupun yang lainnya. Salah satu cara untuk memanipulasi citra adalah dengan melakukan transformasi wavelet.

2.4 Citra Digital

Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada pita magnetik[2]. Menurut presisi yang digunakan untuk menyatakan titik-titik koordinat pada domain spasial atau bidang dan untuk menyatakan nilai keabuaan atau warna suatu citra, maka secara teoritis citra dapat dikelompokkan menjadi empat kelas citra, yaitu citra kontinu-kontinu, kontinu-diskrit, diskrit-kontinu, dan diskrit-diskrit, di mana label pertama menyatakan presisi dari titik-titik koordinat pada bidang citra sedangkan label kedua menyatakan presisi nilai keabuan atau warna. Kontinu dinyatakan dengan presisi angka tak terhingga, sedangkan diskrit dinyatakan dengan presisi angka terhingga

2.6 *Watermarking*

Sesuai dengan penjelasan dalam buku Rinaldi Munir 2006,309 yaitu ada beberapa definisi dari *watermarking,* antara lain *watermarking* berarti suatu konsep untuk menyisipkan suatu data atau pola ke dalam dokumen sehingga suatu potongan informasi seperti kepemilikan atau identitas konsumen yang berhak untuk menggunakannya berada dalam data tersebut. Selain itu *Watermarking* merupakan suatu bentuk dari *Steganography* (Ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain), dalam mempelajari teknik-teknik bagaimana penyimpanan suatu data *digital* ke dalam data *host digital* yang lain (Istilah *host* digunakan untuk data/sinyal digital yang ditumpangi.). namun ada perbedaan antara *watermarking* dan *steganografi*. Jika pada *steganografi* informasi rahasia disembunyikan di dalam media digital dimana media penampung tidak berarti apa-apa, maka pada *watermarking* justru media digital tersebut yang akan dilindungi kepemilikannya dengan pemberian suatu informasi tertentu didalamnya.

2.7 Adaptive Data Hiding untuk Otentikasi Citra Digital

Penyembunyian data sekumpulan besar data secara rapi pada sebuah citra digital tanpa mengakibatkan degradasi yang signifikan, teknik penyembunyian harus beradaptasi pada karakteristik lokal pada sebuah citra [4].

Beberapa ahli menggunakan kriteria global dimana data tersebut disembunyikan, seperti kriteria statistik yang independen dari citra atau kriteria yang cocok dengan citra tertentu. Namun pendekatan ini hanya dapat diterapkan ketika menyembunyikan data berukuran kecil saja.

PEMBAHASAN

3.1 Analisa Masalah

Seiring dengan perkembangan teknologi komputer yang semakin pesat, proses pemberian *copyright* pada produk digital merupakan suatu permasalahan penting yang harus diselesaikan. Proses pemberian *copyright* pada citra digital dapat memanfaatkan teknik *watermarking*.

Teknik *watermarking* yang bagus mampu menyembunyikan sekumpulan besar data secara rapi pada sebuah citra digital tanpa mengakibatkan degradasi yang signifikan. Untuk itu, teknik penyembunyian harus beradaptasi pada karakteristik lokal pada citra. Namun, kebanyakan metode yang dikemukakan hanya mampu menyimpan data berukuran kecil saja.

3.2. Analisa Proses Metode *Adaptive Data Hiding* (ADH)

Prosedur kerja dari skema *Adaptive Data Hiding* untuk Otentikasi Citra Digital dapat digambarkan dalam bentuk *flowchart diagram* seperti berikut:

1. Proses pembuatan *watermark*
   Proses ini memerlukan beberapa data input, seperti citra asli (berupa citra *grayscale*), citra *watermark* (berupa citra hitam putih) dan kunci rahasia (berupa teks / *string*). Proses ini akan menghasilkan sebuah citra *watermark* baru yang merupakan citra hasil gabungan dari citra asli dan citra *watermark* input.
2. Proses ekstraksi *watermark*
   Proses ini memerlukan beberapa input seperti citra asli (citra *grayscale*), citra *watermark* hasil dan input kunci. Sedangkan, output dari proses ini adalah citra *watermark* hitam putih.

Contoh seperti pada tabel berikut:

Tabel 1 Pixel dari citra asli

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Citra Watermark:

1	**1**	**1**	**1**
1	0	0	0
1	0	0	0
1	**1**	**1**	**1**

Kunci = ‘123’.

Citra Watermark dipecah menjadi:

1	**1**
1	0

1	**1**
0	0

1	0
1	**1**

0	0
1	**1**

Citra asli dipecah menjadi :

1. High level partition
   16 x 16 / 4 x 4 = 4 x 4
2. Low level partition
   (4 x 4) x (2 x 2) = (8 x 8)

Tabel 2 Hasil dari citra asli setelah dipeah menjadi (4 x 4) x (2 x 2) = (8 x 8)

255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**

255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**
255	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**	**255**

Hitung nilai :
H(123, 1, 4, 4, 1, 255) = DC5A3C3B
H(123, 2, 4, 4, 2, 255) = BEA5B411
H(123, 3, 4, 4, 3, 255) = F648FE4E
H(123, 4, 4, 4, 4, 255) = D1DB9E43

Misalkan diambil bit paling kanan, maka diperoleh:

Kunci untuk subBlok 1 = B = 1011
Kunci untuk subBlok 2 = 1 = 0001
Kunci untuk subBlok 3 = E = 1110
Kunci untuk subBlok 4 = 3 = 0011

Citra watermark hasil:

1	**1**
1	0

Xor
10
11

0	1
0	1

1	**1**
0	0

Xor
00
01

1	1
0	1

1	0
1	**1**

Xor
11
10

0	1
0	1

0	0
1	**1**

Xor
00
01

0	0
0	0

Jadi citra watermark yang akan ditempelkan :

0	1	1	1
0	1	0	1
0	1	0	0
0	1	0	0

Hasil proses penempelan :

Tabel 3 Hasil penempelan citra *watermark*

254	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
254	255	255	255	255	255	255	255	**254**	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
254	255	255	255	255	255	255	255	**254**	255	255	255	**254**	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
254	255	255	255	255	255	255	255	**254**	255	255	255	**254**	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Proses ekstraksi:
Dari citra *watermarking* yang diperoleh diatas, dibandingkan dengan citra asli, sehingga diperoleh:

0	1	1	1
	1	0	1

0			
0	1	0	0
0	1	0	0

Hitung nilai :
H(123, 1, 4, 4, 1, 255) = DC5A3C3B

H(123, 2, 4, 4, 2, 255) = BEA5B411
H(123, 3, 4, 4, 3, 255) = F648FE4E
H(123, 4, 4, 4, 4, 255) = D1DB9E43

Misalkan diambil bit paling kanan, maka diperoleh:

Kunci untuk subBlok 1 = B = 1011
Kunci untuk subBlok 2 = 1 = 0001
Kunci untuk subBlok 3 = E = 1110
Kunci untuk subBlok 4 = 3 = 0011

XOR kan ulang sehingga diperoleh citra watermark semula.

4. ALGORITMA DAN IMPLEMENTASI

4.1. Algoritma Dengan Metode *Adaptive Data Hiding* (ADH)

Algoritma yang digunakan untuk membuat perangkat lunak dapat dirincikan sebagai berikut:

1. Proses Penempelan

```
Input       :
originalArray ←  warna citra asli (citra grayscale)
originalArray1 ← warna citra hitam putih (citra watermark)
nUkuranX1  ← nilai X dari ukuran blok
nUkuranY1  ← nilai Y dari ukuran blok

Output      :
originalArray2 ← warna citra watermark hasil

Proses      :
Hitung ukuran subblok dari citra watermark.
    nWdt = Original1.Width \ nUkuranX1
    nHgt = Original1.Height \ nUkuranY1
Hitung nilai mean dari setiap subblok.
    nArrMean = 0
    For u = 1 sampai jumlah blok
        For j = 0 To nUkuranY1 - 1
            For i = 0 To nUkuranX1 - 1
                nArrMean = nArrMean + Subblok(u, j, i)
            Next
        Next
        nArrMean = nArrMean / (nUkuranX2 * nUkuranY2)
    Next
Hitung nilai hash dari setiap subblok
    For u = 1 To nWdt * nHgt
        nArrHash(u) = SHA(Kunci + nMean))
    Next
Hitung citra hasil dengan melakukan operasi XOR antara nilai hash dari subblok watermark
    For u = 1 To nWdt * nHgt
        For j = 0 To nUkuranY1 - 1
            For i = 0 To nUkuranX1 - 1
                SubBlokH(u, j, i) = SubBlokWatermark(u, j, i) Xor nArrHash(u)
            Next
        Next
    Next
```

2. Proses Ekstraksi

```
Input       :
originalArray  ←warna citra asli (citra grayscale)
originalArray1  ←warna citra watermark hasil
nUkuranX1 ←nilai X dari ukuran blok
nUkuranY1←nilai Y dari ukuran blok
Output      :
originalArray2  ← warna citra watermark terekstrak
Proses      :
Ekstrak decoded bit dengan membandingkan nilai mean dari subblok watermark
dan subblok citra asli.
    For j = 1 To nSubBlokY1
        For i = 1 To nSubBlokX1
            nMean1 = 0
            For j1 = 0 To nSubBlokY - 1
                For i1 = 0 To nSubBlokX - 1
                    nMean1 = nMean1 + originalArray(j1, i1)
                Next
            Next
            nMean2 = 0
            For j1 = 0 To nSubBlokY - 1
                For i1 = 0 To nSubBlokX - 1
                    nMean2 += originalArray1(j1, i1)
                Next
            Next
            If nMean2 < nMean1 Then
                cDecodedBit(j, i) = 0
            Else
                cDecodedBit(j, i) = 1
            End If
        Next
    Next
Hitung nilai hash dari setiap subblok
    For u = 1 To nWdt * nHgt
        nArrHash(u) = SHA(Kunci + nMean))
    Next
Hitung citra hasil dengan melakukan operasi XOR antara nilai hash dari subblok watermark
    For u = 1 To nWdt * nHgt
        For j = 0 To nUkuranY1 - 1
            For i = 0 To nUkuranX1 - 1
                SubBlokH(u, j, i) = SubBlokWatermark(u, j, i) Xor nArrHash(u)
            Next
```

Next
Next

4.2. Implementasi

Untuk menggunakan perangkat lunak ini, jalankan *file* ”Adaptive Data Hiding.EXE”, maka akan ditampilkan tampilan utama dari program seperti terlihat pada gambar berikut:

Gambar 1 Tampilan Utama

Proses Pembuatan *Watermark*

Untuk melakukan proses pembuatan *watermark*, maka dapat mengklik *link* ‘Proses Pembuatan *Watermark’* sehingga sistem akan menampilkan *form* berikut:

Label16
Label16

Gambar 2 Tampilan Proses Pembuatan *Watermarking* Sebelum Mengeksekusi Proses

Pilihlah citra asli dan citra *watermark* yang ingin diproses. Setelah itu, input data kunci dan ukuran subblok citra *watermark*. Apabila citra input berupa citra berwarna, maka kliklah tombol ‘Convert to Grayscale’ untuk mengkonversikan citra asli ke bentuk *grayscale*.

Citra Asli (Grayscale) : E:\Gambar Foto\City\Antartica.jpg
Citra Watermark (Hitam Putih) : D:\Documents and Settings\ROBIN\Desktop\A.bmp
Kunci : testing
Ukuran subblok citra watermark : 10 X 10
Ukuran citra : 320 x 240
Ukuran citra : 50 x 50

Gambar 3 Tampilan Proses Pembuatan *Watermarking* Setelah Mengeksekusi Proses

Setelah semua data dimasukkan, kliklah tombol ‘Proses’ untuk melakukan proses pembuatan watermark dan melanjutkan ke proses selanjutnya yaitu proses penempelan *watermark*. Sistem akan menampilkan *form* seperti terlihat pada gambar berikut:

Ukuran citra : 320 x 240
Ukuran citra : 50 x 50

Gambar 4 Tampilan Proses Penempelan *Watermarking*

Pada tampilan ini akan ditampilkan citra *watermark* terenkripsi yang akan disisipkan ke dalam citra asli. Setelah itu, kliklah tombol ‘Proses’ sehingga sistem akan menampilkan *form* untuk melakukan proses penempelan *watermark* terenkripsi ke dalam citra *watermark* terenkripsi dan melakukan penyimpanan citra hasil *watermarking*.

Citra Watermark Baru : D:\HASIL.bmp
Ukuran citra : 320 x 240

Gambar 5 Tampilan Proses Penyimpanan Citra Hasil *Watermark*

Proses Ekstraksi Watermark

Untuk melakukan proses pembuatan *watermark*, maka dapat mengklik *link* ‘Proses Ekstraksi Watermark’ sehingga sistem akan menampilkan *form* berikut:

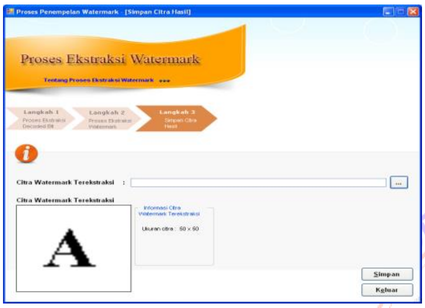
Citra Asli (Grayscale) : E:\Gambar Foto\City\Antartica.jpg
Citra Hasil Watermark : D:\HASIL.bmp
Kunci : testing
Ukuran citra watermark (hitam putih) : 50 X 50
Ukuran subblok citra watermark : 10 X 10
Ukuran citra : 320 x 240
Ukuran citra : 320 x 240

Gambar 6 Tampilan Proses Ekstraksi *Watermarking* untuk Proses Ekstraksi *Decoded Bit*

Setelah itu, kliklah tombol ‘Proses’ sehingga sistem akan menampilkan *form* ‘Ekstraksi Decoded Bit’ seperti terlihat pada gambar berikut:

Gambar 7 Tampilan *Decoded Bit* Hasil Proses Ekstraksi *Watermark*

Setelah itu, kliklah tombol ‘Proses’ sehingga sistem akan menampilkan citra watermark yang terekstrak keluar, seperti terlihat pada gambar berikut:



Gambar 8 Tampilan *Watermark* Hasil Ekstraksi

KESIMPULAN

Setelah menyelesaikan pembuatan perangkat lunak ini, penulis dapat menarik beberapa kesimpulan sebagai berikut:

1. Teknik Otentikasi Citra Digital dengan menggunakan metode *Adaptive Data Hiding* (ADH) dapat diaplikasikan dengan menggunakan *Microsoft Visual Basic* 2008.
2. Dimensi citra input dan citra hasil tidak berubah dan perbedaan warna citra input dan citra hasil juga tidak kelihatan jelas.
3. Skema *Adaptive Data Hiding* untuk Otentikasi Citra Digital mampu mengekstrak keluar citra *watermark*.

DAFTAR PUSTAKA

1. Munir. R, 2006, Pengolahan Citra Digital, Bandung, Informatika Bandung
2. Sutoyo.T dkk, 2009, Teori Pengolahan Citra Digital, Yogyakarta, ANDI dan UDINUS
3. Kadir Abdul,2005, Pengenalan Teknologi Sistem Informasi, Yogyakarta, Andi Offset
4. Sarabjeet S. Bedi, Shekhar Verma and Geetam Tomar Member, IEEE, Citra Digital Untuk Metode Adaptive Data Hiding(ADH) 2010,122.
5. Junindar, 2008 Visual Basic 2008
6. Wikipedia, Table ASCIL, <http://www.ascii.cl/htmlcodes.htm>, 08 Juli 2014, 5:46:10
7. (http://wsilfi.staff.gunadarma.ac.id/Downloads/files/1034/Pengenalan+UML.pdf 08 Juli 2014,09:46:10)