RANCANG BANGUN APLIKASI WATERMARKING PADA GAMBAR DENGAN ALGORITMA DIGITAL SEMIPUBLIC

Ng Poi Wong, Willy Cahyadi

STMIK Mikroskil
Jl. Thamrin No. 122, 124, 140 Medan 20212

poiwong@mikroskil.ac.id

Abstrak

Sasaran *watermarking* adalah untuk menyembunyikan pesan (*cover message*). Pada skema *public watermarking*, tidak memungkinkan seseorang untuk mendeteksi *watermarking* apabila telah terjadi pergantian terhadap *watermarking*. Selain itu, pada skema *public watermarking, cover message* tidak dapat di-*recover* kembali. Sedangkan, pada skema *private watermarking*, untuk dapat mendeteksi *watermarking*, maka seseorang harus mengetahui kunci privat yang bersesuaian. Kedua skema tersebut memiliki kelebihan dan kelemahannya masing-masing. Skema *watermarking* (WM) yang bersifat semi publik, dapat digunakan kunci privat untuk mendeteksi *watermark* apabila telah terjadi penyerangan terhadap *watermarking* dan kunci privat dapat digunakan mengekstrak kembali *cover message*.
Proses pembentukan *watermarking* dimulai dari proses pembacaan piksel citra, kemudian dilakukan perhitungan nilai *stegomessage* yang akan disisipkan dan proses penyisipan semua nilai yang diperlukan dalam proses pengecekan *watermarking*. Apabila terjadi penyerangan, maka pada saat pembentukan *watermarking*, akan dilakukan perhitungan nilai noise tertentu (secara acak) yang akan disisipkan ke dalam citra. Setelah proses pembentukan *watermarking*, maka dilanjutkan dengan proses pengecekan *watermarking* dan proses ekstraksi pesan.
Aplikasi ini mampu menghasilkan *watermarking* yang disisipkan pada citra input dan proses pengecekan *watermarking* yang telah disisipkan. Selain itu juga mampu menghasilkan proses perhitungan dari tahapan pembentukan dan pengecekan *watermarking*, serta tersedia juga fasilitas untuk melakukan penyerangan terhadap skema.

***Kata kunci** : watermarking, kunci public, cover message, data digital*

1. Pendahuluan

Penyembunyian informasi atau *information hiding* (IH) merupakan sebuah area yang masih baru dalam sekuritas informasi. Sasaran dari desain IH adalah untuk menyembunyikan pesan walaupun pada kenyataannya dalam pesan mungkin terdapat data yang tidak berguna, biasanya disebut sebagai *cover message* (CM) [1]. *Watermarking* (WM) merupakan salah satu aplikasi IH. CM harus dikombinasikan dengan beberapa informasi lainnya, seperti identifikasi pemilik. Kemudian kode identifikasi adalah ditempelkan secara permanen pada data dan harus tetap tersedia diantara data setelah sembarang proses transformasi yang dilakukan oleh penyerang yang dilakukan untuk membuang WM, menjaga kualitas dari CM [1][2].

Pada tahun 2002, Valery Korzhik dan Guillermo Morales-Luna memperkenalkan sebuah skema *Digital Semipublic Watermarking* dimana diambil asumsi bahwa hanya terdapat satu tipe serangan yaitu *additive noise attack*. Terdapat dua tipe utama dari WM yaitu versi privat, dimana *encoder* dan *decoder* menggunakan kunci rahasia, dan versi publik (versi *blind*), dimana tidak ada informasi apapun yang tersedia untuk *decoder*. Karena bentuk semiprivat telah tersedia, maka Valery Korzhik dan Guillermo Morales-Luna memperkenalkan bentuk semipublik dari WM [3]. Pada kasus ini, diasumsikan bahwa setiap *user* mampu untuk

mengekstraksi WM tanpa kunci rahasia. Pada waktu yang sama, pembuat WM, memiliki kunci rahasia, mampu untuk mendeteksi WM, bahkan setelah sebuah penyerangan.

Yang menjadi permasalahan dari penulisan ini adalah skema *watermarking* (WM) yang bersifat semi publik, dapat digunakan kunci privat untuk mendeteksi *watermark* apabila telah terjadi penyerangan terhadap *watermarking*. Untuk menerapkan skema tersebut maka perlu dirancang sebuah aplikasi yang menerapkan skema *Semi-Public Watermarking*.

Tujuan dari penulisan ini adalah untuk menerapkan algoritma *Digital Semipublic Watermarking* untuk melakukan pengamanan data dan menyisipkannya ke dalam suatu gambar. Sedangkan manfaat dari penulisan ini adalah diharapkan dapat digunakan sebagai referensi dalam membantu memberikan gambaran dan pemahaman mengenai algoritma *Digital Semipublic Watermarking*, serta mendeskripsikan contoh dari penggunaan *Digital Semipublic Watermarking*.

Dalam membangun aplikasi ini dilakukan pembatasan untuk citra JPG, BMP dan GIF tanpa animasi, panjang dari pesan maksimal 100 karakter, kunci yang digunakan terdiri dari kunci publik dan kunci privat (kunci dapat di-input secara manual ataupun dihasilkan secara acak oleh aplikasi), kunci privat berupa deretan nilai +1 dan -1, jenis penyerangan yang dilakukan adalah penyerangan untuk mengacaukan watermarking, dan data kunci yang digunakan dalam penyerangan akan dihasilkan secara acak oleh aplikasi.

2. Kajian Pustaka

2.1 Citra Digital

Citra adalah representasi dari sebuah objek. Citra merupakan kumpulan dari titik-titik yang mempunyai intensitas tertentu membentuk satu kesatuan perpaduan yang mempunyai arti baik secara artistik maupun intristik. Citra *Digital* merupakan suatu array dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar. Jadi informasi yang terkandung bersifat diskrit [4].

Salah satu sistem yang digunakan untuk mewakili gambar yaitu sistem warna RGB (*Red*, *Green*, *Blue*). Sistem RGB adalah sistem yang menggabungkan warna primer gabungan (*additive primary colours*) untuk memperoleh gabungan warna [5].

Untuk menyimpan foto dan citra digunakan format citra layar kuadratis yang terdiri atas titik-titik citra kecil yang disebut dengan piksel (*pixel*) atau dot. Piksel berbentuk bujur sangkar dengan ukuran relatif kecil. Banyaknya piksel tiap satuan luas tergantung pada resolusi yang digunakan. Keanekaragaman warna piksel tergantung pada *bit depth* yang dipakai. Semakin banyak jumlah piksel tiap satu satuan luas, semakin baik kualitas citra yang dihasilkan dan tentu akan semakin besar ukuran filenya [5][7].

Resolusi adalah jumlah piksel per satuan luas yang ada suatu citra. Satuan piksel yang sering dipakai adalah dpi (*dot per inch*) atau ppi (piksel per inch). Satuan dpi menentukan jumlah piksel yang ada setiap satu satuan luas. Dalam hal ini adalah satu inch kuadrat. Resolusi sangat berpengaruh pada detil dan perhitungan citranya [6][7].

Bit Depth (kedalaman warna) sering disebut dengan *pixel depth* atau *color depth*. *Bit Depth* menentukan berapa banyak informasi warna yang tersedia untuk ditampilkan dalam setiap piksel. Semakin besar nilainya semakin bagus kualitas citra yang dihasilkan. Tentu ukurannya juga semakin besar [5][7].

2.2 Watermarking

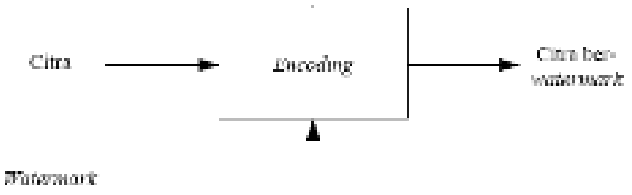
Watermarking adalah suatu konsep untuk menyisipkan suatu data atau pola ke dalam dokumen sehingga suatu potongan informasi seperti kepemilikan atau identitas konsumen yang berhak untuk menggunakannya berada dalam data tersebut [2]. Definisi lain, *Watermarking* merupakan suatu cara untuk penyembunyian atau penanaman data/informasi tertentu, baik hanya berupa catatan umum maupun rahasia ke dalam suatu data *digital* lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia, khususnya indera penglihatan atau pendengaran, dan mampu menghadapi proses-proses pengolahan sinyal *digital* sampai pada tahap tertentu [9]. Selain itu *Watermarking* merupakan suatu bentuk dari *Steganography* yaitu ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain [1]. 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 [8].

2.2.1 Digital Watermarking

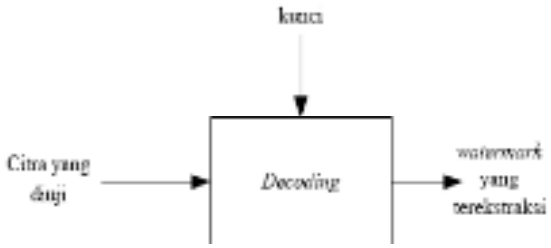
Digital Watermarking adalah penyisipan sinyal *digital* ke dalam suatu media *digital*. *Digital Watermarking* dimulai dari proses-proses pengolahan sinyal *digital*, dimana sinyal *digital* dapat berupa gambar, *audio*, *video*, dan teks [5]. *Digital Watermarking* dapat diimplementasikan dengan memanfaatkan kekurangan dari indera manusia terutama penglihatan dan pendengaran, dimana indera tersebut kurang sensitif terhadap perubahan yang terjadi, misalnya perubahan pada level bit, atau perubahan pada level frekuensi [9]. Hal yang memisahkan *watermarking* dari steganografi adalah dalam implementasinya, steganografi digunakan untuk mengamankan informasi yang ditumpangkan pada suatu media *digital*, sedangkan *watermarking* dapat dimanfaatkan untuk berbagai tujuan misalnya sebagai indikator perubahan pada data yang di-*watermarking* (*Tamper-Proofing*), sebagai alat identifikasi isi data *digital* pada lokasi tertentu (*Feature Location*), sebagai keterangan tentang data *digital* itu sendiri atau informasi lain yang dipandang perlu untuk ditanamkan kedalam media yang bersangkutan (*Annotation/Caption*), komunikasi yang aman (*Secure and Invisible Communications*), sebagai metode untuk penyembunyian label hak cipta pada data *digital* sebagai bukti otentik kepemilikan karya *digital* (*Copyright-Labeling*), sebagai *watermark* yang terlihat oleh indera manusia (*Visible Watermarking*) dan *watermark* tidak tampak (*Invisible Watermarking*), serta sebagai proses verifikasi *watermark* tanpa citra asal (*Blind Watermarking*) dan proses verifikasi *watermark* dengan citra asal (*Non Blind Watermarking*) [8][9].

2.2.2 Framework Watermarking

Jika *watermark* merupakan sesuatu yang ditanamkan, maka *watermarking* merupakan proses penanaman *watermark* tersebut. Secara umum *framework* sebuah algoritma *watermarking* tersusun atas dua bagian, yaitu algoritma penyisipan *watermark* (*encoder*) dan algoritma pendeteksian *watermark* (*decoder*). Algoritma penyisipan *watermark* (*encoder*) adalah algoritma yang menangani bagaimana sebuah *watermark* ditanamkan pada media induknya, sedangkan algoritma pendeteksian *watermark* adalah algoritma yang menentukan apakah di dalam sebuah media *digital* terdeteksi *watermark* yang sesuai atau tidak [2][8].



Gambar 1. Proses penyisipan *watermark* pada citra *digital* [9]



Gambar 2. Proses pendeteksian *watermark* pada citra *digital* [9]

Label *watermark* adalah suatu data/informasi yang akan dimasukkan ke dalam data *digital* yang ingin di-*watermark*. Ada 2 jenis label yang dapat digunakan, yakni teks biasa, dimana label *watermark* dari teks tersebut biasanya menggunakan nilai-nilai ASCII dari masing-masing karakter dalam teks yang kemudian dipecahkan atas bit per bit. Kelemahan dari label jenis ini adalah kesalahan pada satu bit saja dapat menghasilkan hasil yang berbeda dengan teks asli. Sedangkan jenis berikutnya adalah logo atau citra atau suara, dimana berbeda dengan teks yakni kesalahan pada beberapa bit masih dapat memberikan persepsi yang sama dengan aslinya oleh pendengaran maupun penglihatan kita, tetapi kerugiannya adalah jumlah data yang cukup besar [9].

2.3 Digital Semipublic Watermarking

Pada kasus *Stegomessage* (SM) atau dengan perkataan lain *Watermarked Message* memiliki bentuk seperti persamaan (1) berikut [9] :

$$S(n) = C(n) + w(n), \qquad n = 1, 2, \ldots, N \qquad (1)$$

dimana C(n) adalah *cover message* (CM), $w(n) = \Delta(n)e(n)$, $\Delta(n)$ adalah sebuah deretan nilai real non-negatif, e(n) adalah sebuah deretan dari tanda + 1, - 1 dan N adalah jumlah elemen WM seperti piksel citra CM.

Pada versi *semipublic* dari WM, deretan e(n) diasumsikan diketahui untuk setiap user dan merupakan kunci publik, sedangkan deretan $\Delta(n)$ disimpan secara rahasia dan merupakan kunci privat. Deretan e(n) menggunakan sampel yang bebas dan terdistribusi secara teratur dari tanda ± 1 dan $\Delta(n)$ juga mengambil sampel non negatif yang bebas dan terdistribusi secara teratur dalam sebuah interval (0, D), dimana $D > 0$ adalah sebuah nilai positif yang tetap. *Cover message* C(n) dideskripsikan sebagai sebuah proses *discrete zero-mean* acak dengan *variance* σ^2_C. Batasan distorsi setelah *watermarking* diberikan sebagai sebuah *signal-to-noise ratio* yang dapat dideskripsikan seperti terlihat pada persamaan (2) berikut [9]:

$$\eta_\omega = \frac{var\,(C(n))}{var\,(\Delta(n)e(n))} = \frac{3\sigma_C^2}{D^2} \qquad (2)$$

Valery Korzhik, Guillermo Morales-Luna, Dmitry Marakov dan Irina Marakova juga memperkenalkan untuk setiap *user* biasa untuk mendeteksi WM dapat menggunakan rumusan (3) berikut [9]:

$$\Lambda = \sum_{n=1}^{N} S(n)e(n) \qquad (3)$$

Jika $\Lambda \geq \lambda$, maka sebuah WM telah terdeteksi, jika tidak, maka WM tidak terdeteksi. Jika WM terdapat dalam SM, maka diperoleh rumusan (4) berikut [9]:

$$\Lambda = \sum_{n=1}^{N} \big(C(n) + \Delta(n)e(n)\big)e(n) =: \Lambda_1 \qquad (4)$$

Jika tidak terdapat dalam SM, maka diperoleh rumusan (5) berikut [9]:

$$\Lambda = \sum_{n=1}^{N} C(n)e(n) =: \Lambda_0 \quad (5)$$

Sekarang, anggap sebuah penyerangan dilakukan oleh *user* biasa. Keputusan mengenai keberadaan atau tidak dari WM diambil setelah membandingkan nilai berikut dengan sebuah *threshold* λ seperti terlihat pada rumusan (6) berikut [9]:

$$\Lambda' = \sum_{n=1}^{N} S'(n)e(n) \quad (6)$$

Dimana S'(n) = S(n) + ε(n), untuk sebuah *attacking additive noise* ε(n). Karena penyerang mengetahui *sequence* ε(n), untuk n = 1, …, N, maka dia dapat mendeteksi keberadaan atau tidak dari WM pada CM, sehingga dia dapat membuat sebuah deretan *additive noise* ε(n), untuk n = 1, …, N, seperti terlihat pada rumusan (7) berikut [9]:

$$\varepsilon(n) = \begin{cases} \sigma e(n) & if\ WM\ is\ absent \\ 0 & otherwise \end{cases} \quad (7)$$

Oleh karena itu, untuk seorang *user* biasa yang mencoba untuk mendeteksi WM setelah penyerangan, terdapat dua kemungkinan hasil, seperti terlihat pada rumusan (8) berikut [9]:

$$E(\Lambda') = \begin{cases} E(\Lambda'_0) = N.\sigma \\ E(\Lambda'_1) = \frac{N.D}{2} \end{cases} \quad \begin{matrix} if\ WM\ is\ absent \\ otherwise \end{matrix} \quad (8)$$

Berdasarkan hasil pengujian yang dilakukan oleh pengembang algoritma, diketahui bahwa algoritma diatas tidak dapat diterapkan secara praktikal, dan mereka menganjurkan untuk menerapkan metode *semipublic* WM lainnya yaitu yang berdasarkan pada deretan periodik. Anggap SM didefinisikan seperti terlihat pada rumusan (9) berikut [9]:

$S(n) = C(n) + w(n)$, dimana $n = 1, 2, \ldots, 2N_0$ (9)

Dimana:

$w(n) = \alpha\pi(n)$ (10)

$N_0 = N / 2$ (11)

Deretan π(n) adalah deretan acak yang terdiri dari tanda ±1, periodik dengan dua periode dengan panjang masing-masing N_0. Nilai *threshold* dapat dihitung dengan menggunakan rumusan (12) berikut [9]:

$$\Lambda = \sum_{n=1}^{N_0} S(n)S(n + N_0) \quad (12)$$

Sementara itu, untuk menambahkan *noise*, maka dapat digunakan rumusan (13) dan rumusan (14) berikut [9] :

1. Untuk kasus tidak ada WM:

$$\epsilon(n) = \begin{cases} \sigma\tilde{\pi}(n) & n \le N_0 \\ \sigma\tilde{\pi}(n - N_0) & N_0 + 1 \le n \le 2N_0 \end{cases} \quad (13)$$

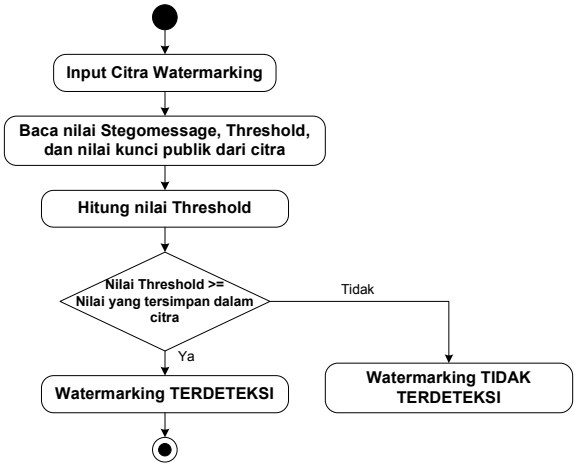
2. Untuk kasus ada WM:

$$\epsilon(n) = \begin{cases} \sigma\tilde{\pi}(n) & n \le N_0 \\ -\sigma\tilde{\pi}(n - N_0) & N_0 + 1 \le n \le 2N_0 \end{cases} \quad (14)$$

3. Metode Penelitian

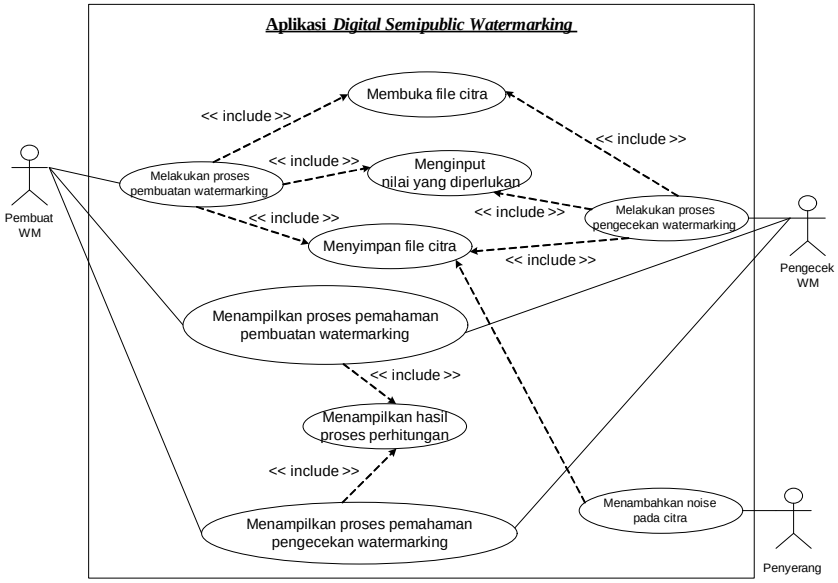
Digital Semipublic Watermarking merupakan suatu konsep untuk menyisipkan suatu data ke dalam citra sehingga suatu potongan informasi seperti kepemilikan atau identitas konsumen yang berhak untuk menggunakannya berada dalam citra tersebut. Dalam proses kerja *Digital Semipublic Watermarking*, terdapat dua buah proses yaitu proses pembuatan dan pengecekan *watermarking*. Secara garis besar, proses pembuatan dan pengecekan *watermarking* dapat diilustrasikan pada gambar 3 dan gambar 4.

Gambar 3. Proses pembuatan *watermarking*



Gambar 4. Proses pengecekan *watermarking*

Aplikasi yang dirancang digambarkan dan dimodelkan dengan menggunakan *use case* pada gambar 5 berikut:



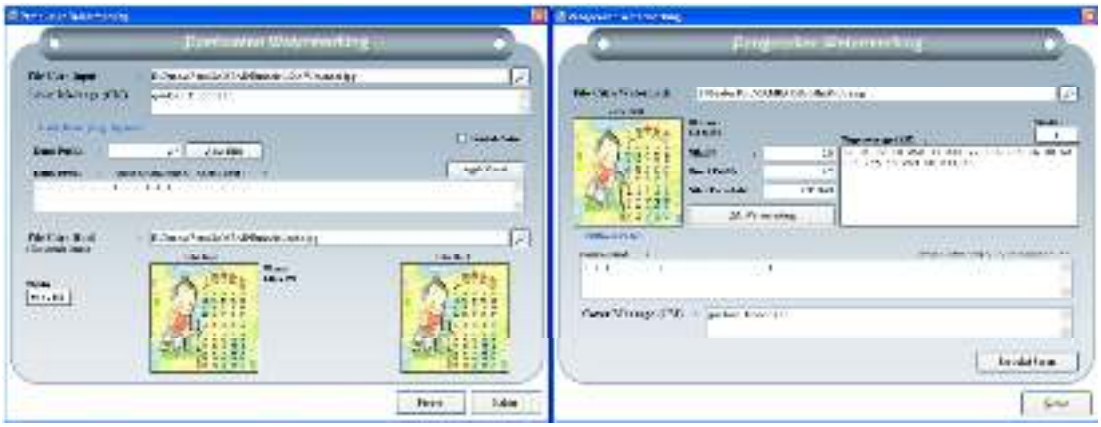
Gambar 5. *Use case* Aplikasi *Digital Semipublic Watermarking*

Seperti terlihat pada gambar 5 di atas, entitas dari sistem hanya berjumlah dua buah yaitu pembuat *watermarking*, pengecek *watermarking* dan penyerang. Sedangkan, proses yang terdapat pada perangkat lunak ada 8 buah yaitu melakukan proses pembuatan *watermarking*, melakukan proses pengecekan *watermarking*, membuka file citra, menyimpan file citra, menginput nilai yang diperlukan, menampilkan proses pemahaman pembuatan *watermarking*, menampilkan proses pemahaman pengecekan *watermarking* dan menampilkan hasil proses perhitungan.

4. Hasil dan Pembahasan

4.1 Hasil Tampilan

Aplikasi yang telah dirancang dibangun untuk menerapkan *Digital Semipublic Watermarking* pada gambar memiliki beberapa tampilan form, yakni form pembuatan *watermarking* dan form pengecekan *watermarking* seperti gambar 6, serta form pemahaman proses pembuatan *watermarking* dan form pemahaman proses pengecekan *watermarking* seperti gambar 7.



Gambar 6. Form pembuatan dan pengecekan *watermarking*

Gambar 7. Form pemahaman proses pembuatan dan pengecekan *watermarking*

Pada form pembuatan *watermarking*, pengguna aplikasi dapat menginput file gambar awal yang belum di-*watermark*, kemudian pengguna dapat menginput *cover message* yang ingin disisipkan ke dalam file gambar tersebut dengan parameter nilai kunci privat dan kunci publik yang juga diinput oleh pengguna. Sedangkan pada form pengecekan *watermarking*, pengguna dapat menginput file gambar yang telah di-*watermark*, dan menginput nilai kunci privatnya.

Pada form pemahaman proses pembuatan dan pengecekan *watermarking*, hal yang sama dengan form pembuatan dan pengecekan *watermarking*, pengguna diwajibkan menginput file gambar dan beberapa nilai yang diperlukan. Bedanya terdapat fitur penjabaran bagaimana proses pembuatan dan pengecekan *watermarking* tersebut dilakukan secara tahap demi tahap, sehingga pengguna dapat memahami bagaimana proses tersebut dilakukan.

4.2 Hasil Pengujian

Dengan menggunakan aplikasi *Digital Semipublic Watermarking* tersebut, dilakukan pengujian ukuran file yang telah di-*watermark* terhadap beberapa variabel, yakni format file gambar yang berbeda, panjang ukuran *cover message*, dan pemberian *noise*. Pengujian yang dilakukan terhadap format file gambar adalah dilakukan pada format file JPG, BMP, dan GIF. Sedangkan untuk panjang ukuran *cover message* dilakukan dengan menyisipkan pesan dengan ukuran yang berbeda-beda. Kemudian untuk pemberian *noise*, akan dilakukan pengeditan terhadap nilai RGB dari piksel dan nilai *Hue, Saturation*, dan *Lightness* dari file gambar yang sudah di-*watermark*.

Pada pengujian terhadap format file gambar JPG, BMP, dan GIF, perubahan drastis terhadap ukuran file yang telah di-*watermark* terjadi pada format file JPG, dimana berdasarkan pengujian diperoleh bahwa ukuran dari file hasil *watermark* meningkat berkisar 500-750% dari ukuran file awal sebelum di-*watermark*. Sedangkan untuk format file GIF, ukuran file hasil *watermark* meningkat berkisar 25-75%, dan pada format file BMP, peningkatan ukuran file hasil *watermark* relatif sangat kecil yakni berkisar 2-10%.

Pada pengujian terhadap *cover message* dengan panjang yang berbeda-beda, diperoleh bahwa ukuran panjang pesan *cover message* tidak mempengaruhi ukuran file hasil *watermark*.

Pada pengujian terhadap penambahan *noise*, dilakukan perubahan nilai RBG dari beberapa piksel dari file hasil *watermark* dengan menggunakan aplikasi *Adobe Photoshop*. Berdasarkan hasil pengujian, *watermark* tetap berhasil dideteksi, akan tetapi hasil *cover message* yang berhasil di-*decode* menjadi tidak sesuai dengan *cover message* aslinya.

Kemudian penambahan *noise* juga dilakukan dengan mengubah nilai *Hue, Saturation*, dan *Lightness* dari file hasil *watermark*, sehingga berdasarkan hasil pengujian, *watermark* tidak dapat dideteksi dan *cover message* gagal di-*decode*.

5. Kesimpulan

Setelah menyelesaikan perancangan dan pembuatan aplikasi, serta pengujian terhadap *Digital Semipublic Watermark* ini, maka dapat ditarik kesimpulan sebagai berikut :

1. Berdasarkan hasil pengujian yang dilakukan, diperoleh informasi bahwa untuk citra berformat *.JPG, ukuran citra hasil watermarking akan bertambah besar sedangkan untuk citra berformat *.GIF dan *.BMP, perubahan ukuran citra hasil watermarking relatif kecil.
2. Algoritma *Digital Semipublic Watermarking* dapat digunakan untuk menambahkan *watermarking* pada citra dan mampu untuk mendeteksi citra walaupun telah terjadi penyerangan pada citra *watermarking*.

Referensi

[1] Bandyopadhyay, Samir K., 2008, *A Tutorial Review on Steganography*, University of Calcutta.

[2] Mohanty, S. P., 1999, *Digital Watermarking : A Tutorial Review*, Department of Computer Science and Engineering, University of South Florida.

[3] Valery K., G. Morales-Luna, D. Marakov dan I. Marakova, 2002, *Digital Semi-Public Watermarking*, Informatica 26.

[4] Murni, A., 1992, *Pengantar Pengolahan Citra*, PT. Elex Media Komputindo, Jakarta.

[5] Gonzales, R. C. dkk, 1992, *Digital Image Processing*, Addison-Wesley Publishing Company.

[6] Nalwan, A., 1997, *Pengolahan Gambar Secara Digital*, PT. Elex Media Komputindo, Jakarta.

[7] Pratisto, A. S., 2003, *Format Optimal Untuk Setiap Gambar*, CHIP Juni.

[8] Duan, F. Y., I. King, 1999, *A Short Summary of Digital Watermarking Techniques for Multimedia Data*, Proceedings of the 1999, Hong Kong.

[9] Yullinda, C.D., 2008, *Implementasi Watermarking dengan Metode Discrete Cosine Transform (DCT) pada Citra Digital*, Digital Library-Perpustakaan Pusat UNIKOM.