Self Embedding Watermarking pada Citra Digital

Leonardo Refialy

Corresponding author : leo.refialy@*gmail.com*
Universitas Kristen Indonesia Maluku
Jl. Ot Pattimaipauw, Talake Kecamatan Nusaniwe, Kel Wainitu, Kota Ambon, 97155- Indonesia

Abstract-- *Digital watermark is a sign embedded in the digital media, such as images, audio, or video. Digital watermarks typically used to identify ownership of the copyright of digital media. Digital watermarks may be used to verify the authenticity or integrity of the carrier media. Self-embedding means insert a duplicate of a digital image, into itself. Self-embedding can be implemented as watermarking. This modification will provide the advantage that the introduction of area changes / media damage if it fails to verify the authenticity of the digital media. Duplicates of the digital image is made, by choosing 2 MSB, using selective Bitplane, then pasted on the LSB digital image itself. In this study produced a self-embedding application, which can detect changes in the digital image. The results show that the technique of self-embedding and selective bitplane can be used to verify authentication or integrity, and also the detection of the location of the damage on a 24-bit color digital image.*

Keywords: *Digital Watermark, Self-Embedding*.

Abstrak-- Digital watermark adalah suatu tanda yang disisipkan dalam media digital, yaitu gambar, audio, atau video. Digital watermark pada umumnya digunakan untuk tujuan mengidentifikasi kepemilikan media digital tersebut. Digital watermark juga dapat digunakan untuk verifikasi otentikasi atau integritas dari media yang diberi tanda. Self-embedding berarti menyisipkan duplikat suatu citra digital, ke dalam citra digital itu sendiri. Self-embedding dapat diimplementasikan sebagai watermarking. Kombinasi ini akan memberikan keuntungan yaitu pengenalan area perubahan/kerusakan media jika ternyata verifikasi keaslian media digital gagal. Duplikat dari citra digital tersebut dibuat, dengan memilih 2 MSB, menggunakan metode selective bitplane, kemudian disisipkan pada LSB citra digital itu sendiri. Pada penelitian ini dihasilkan sebuah aplikasi self-embedding, yang dapat mendeteksi perubahan pada citra digital. Hasil pengujian menunjukkan bahwa teknik self-embedding dan selective bitplane dapat digunakan untuk verifikasi otentikasi atau integritas, dan juga deteksi lokasi kerusakan pada suatu citra digital 24 bit warna.

Kata kunci : Digital Watermark, Self-Embedding.

PENDAHULUAN

Digital watermark adalah suatu tanda yang disisipkan dalam media digital, yaitu gambar, audio, atau video.[1] Digital watermark pada umumnya digunakan untuk tujuan mengidentifikasi kepemilikan media digital tersebut.[2] Digital watermark juga dapat digunakan untuk verifikasi otentikasi atau integritas dari media yang diberi tanda. [3]

Watermarking

Watermarking merupakan suatu bentuk dari *steganography*, yaitu ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain. *Watermarking* terbagai menjadi dua, yaitu: *visible* dan *invisible*. Jadi *invisible watermarking* merupakan suatu cara untuk menyembunyikan atau menanam suatu data/informasi tertentu ke dalam suatu data digital lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia.[1] *Watermarking* diartikan sebagai suatu teknik penyembunyian data atau informasi "rahasia" kedalam suatu data lainnya untuk "ditumpangi" (kadang disebut dengan *host* data), tetapi orang lain tidak menyadari kehadiran adanya data tambahan pada *host*. Jadi seolah-olah tidak ada perbedaan antara data *host* sebelum dan sesudah proses *watermarking*. *Watermark* harus tahan terhadap berbagai jenis pengolahan/proses digital yang dilakukan pada *host.*[6]

Self Embedding

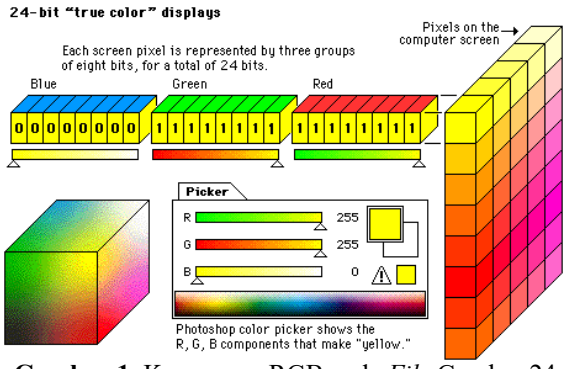
Self-embedding berarti menyisipkan duplikat suatu citra digital, ke dalam citra digital itu sendiri. *Self-embedding* dapat diimplementasikan sebagai *digital watermarking.*[5]

Kombinasi ini akan memberikan keuntungan yaitu pengenalan area perubahan/kerusakan media jika ternyata verifikasi keaslian media digital gagal. Duplikat dari citra digital tersebut dibuat, dengan memilih 2 MSB, menggunakan metode *selective bitplane*, kemudian disisipkan pada LSB citra digital itu sendiri. *Self-embedding* bekerja dengan cara mengambil salinan dari gambar asli. Salinan ini memiliki ukuran yang lebih kecil dari gambar asli. Untuk membuat salinan ini dapat digunakan beberapa teknik seperti kompresi, atau *partial-selection*. Dalam teknik kompresi, gambar dimampatkan sampai memiliki ukuran yang dapat disisipkan ke dalam gambar asli. Dalam teknik partial-selection, sebagian *bit* dari gambar asli dibaca kemudian disisipkan ke dalam gambar tersebut.[7] Teknik penyisipan juga dapat digunakan beberapa teknik, salah satunya adalah LSB. Dengan menggabungkan teknik *selective bitplane* dengan LSB *embedding*, berarti mengambil sebagian MSB dari gambar asli (menjadi gambar salinan), kemudian gambar salinan tersebut disisipkan sebagai *bit* LSB pada gambar asli.[8] Sebuah metode otentikasi yang efektif memiliki beberapa kriteria sebagai berikut: (1) Dapat mengetahui apakah sebuah gambar telah mengalami perubahan atau tidak; (2) Dapat menunjukkan lokasi perubahan yang terjadi; (3) Dapat mengintegrasikan (menggabungkan) antara data otentikasi dengan citra digital yang akan diamankan; (4) Data otentikasi yang disisipkan tidak terlihat dalam kondisi normal (tersembunyi); (5) Data otentikasi tersimpan dalam format *lossy-compression*.[9]

Pada penelitian ini dihasilkan sebuah aplikasi *self-embedding*, yang dapat mendeteksi perubahan pada citra digital. Hasil pengujian menunjukkan bahwa teknik *self-embedding* dan *selective bitplane* dapat digunakan untuk verifikasi otentikasi atau integritas, dan juga deteksi lokasi kerusakan pada suatu citra digital 24 *bit* warna.

Digital Watermark

Digital watermark pada penelitian ini diimplementasikan pada citra digital 24 bit warna (true colour image). Suatu *true colour image* memiliki komponen *red, green* dan *blue* yang terpisah untuk tiap pikselnya. Pada sebagian besar *truecolourimage*, tiap komponen diwakilkan dengan satu *byte* yang terdiri dari 8 *bit*, sehingga setiap piksel memiliki 24 *bit* informasi warna. Oleh karena itu, mode ini sering disebut sebagai "24-*bit* warna". Pada Gambar 1 ditunjukkan contoh potongan dari *file* gambar**.** Potongan ini memiliki dimensi 6 x 6 piksel.Pada tiap piksel terdapat tiga bagian warna yaitu *red, green* dan *blue*. Pada lokasi piksel 1,1 terdapat warna *red* bernilai 96, *green* bernilai 143, dan *blue* bernilai 179. Pada lokasi piksel 1,2 terdapat warna *red* 61, *green* 125, dan *green* 198.[10]



Gambar 1. Komponen RGB pada *File* Gambar 24 *Bit* Warna

METODE PENELITIAN

Tahapan Penelitian

Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam empat tahapan, yaitu: (1) Identifikasi Masalah, (2) Perancangan sistem, (3) Implementasi sistem, dan (4) Pengujian sistem dan analisis hasil pengujian.

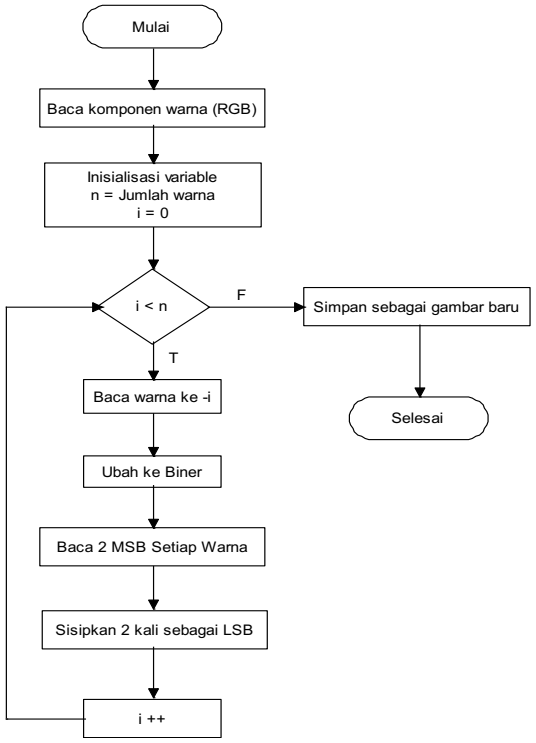
Tahap pertama: mengidentifikasi masalah dan pengumpulan data, adalah bagaimana sistem *Self Embedding* yang dibuat dapat melakukan proses penyisipan dan ekstraksi data gambar dengan metode *selective bitplane*.

Tahap kedua: merancang sistem yaitu proses *embedding* dan *Extraction* menggunakan algoritma *Selective*

Bitplane dan *Self Embedding* yang terjadi di dalam sistem, serta rancangan antarmuka aplikasi yang digunakan oleh *user*. Perancangan proses terdiri dari perancangan proses *Selective Bitplane* yaitu proses pemilihan lokasi *bit*, dan perancangan proses *embedding* dan *extraction*

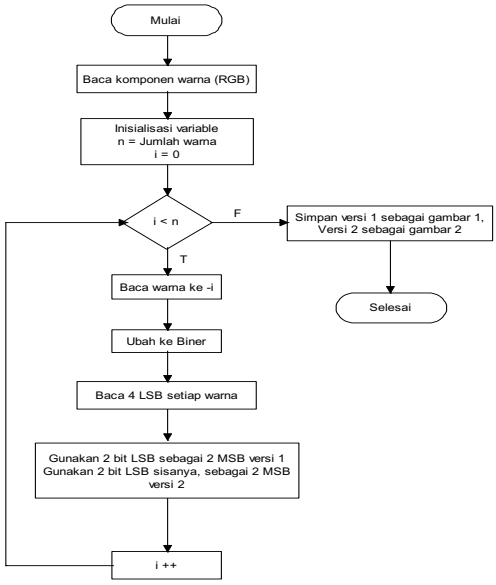
Tahap ketiga: yaitu mengimplementasikan hasil perancangan sistem, membangun sistem berdasarkan rancangan pada tahap sebelumnya. Sistem dikembangkan dalam bentuk aplikasi dekstop. Citra yang digunakan memiliki format BMP atau PNG.

Tahap keempat: adalah melakukan pengujian sistem dan kemudian melakukan analisis terhadap hasil pengujian tersebut. Pengujian dilakukan untuk mengetahui apakah tujuan pembuatan sistem telah tercapai yaitu berhasil menyisipkan salinan dari gambar tersebut ke dalam gambar itu sendiri, kemudian melakukan ekstraksi dari gambar tersebut.

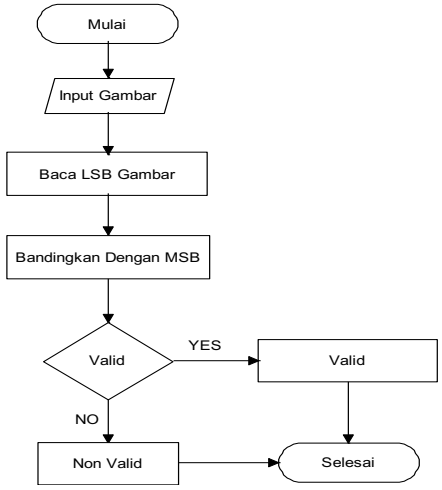


Gambar 2. Proses Penyisipan Digital Watermark

Untuk mengetahui perubahan pada citra digital, disisipkan data otentikasi berupa duplikat dari citra digital itu sendiri (*Host Image*). Duplikat diperoleh dari 2 *bit* MSB tiap warna dari citra digital. 2 *bit* MSB ini kemudian disisipkan dua kali sebagai Versi 1 dan Versi 2. Versi 2 disisipkan pada posisi setelah Versi 1. Hal ini memberikan keuntungan yaitu jika citra digital mengalami perubahan, maka salah satu atau kedua versi duplikat juga akan mengalami perubahan. Namun, karena Versi 1 dan Versi 2 tidak disisipkan dilokasi yang sama, maka perubahan yang terjadi akan berbeda antara kedua versi tersebut. Hal ini ditunjukkan pada pengujian



Gambar 3. Proses Ekstraksi Digital Watermark



Gambar 4. Proses Deteksi Perubahan pada Gambar

Proses embedding dan extraction ditunjukkan pada Gambar 2 dan Gambar 3. Proses deteksi perubahan ditunjukkan pada Gambar 4.

Contoh proses *embedding* dan ekstraksi adalah sebagai berikut: Jika pada proses *embedding* diketahui suatu gambar dengan susunan dua piksel pertama yaitu RGB(125,250,202) dan RGB(180,80,10), maka langkah awal yang dilakukan adalah mengubah nilai tiap warna menjadi biner.

	Piksel			Piksel		
Komponen warna	R	G	B	R	G	B
Nilai Warna	125	250	202	180	80	10
Nilai Warna dalam biner	01111101	11111010	11001010	10110100	01010000	00001010

Gambar 5. Proses Embedding Langkah 1

Langkah selanjutnya adalah membaca 2 bitplane MSB dari tiap*bit* warna tersebut. 2 bitplane MSB dipilih karena sejumlah 2 *bit* tersebut sudah dapat mewakili salinan suatu *file* gambar. Pada gambar 6, terdapat 12

kumpulan *bit*.

MSB 2 bit	01	11	11	10	01	00

Gambar 6. Proses Embedding Langkah 2

Langkah terakhir adalah menggandakan tiap*bit* MSB, sehingga jumlahnya menjadi 2 kali lebih banyak. Pada Gambar 7, terdapat 2 x 12 *bit* = 24 *bit* yang akan disisipkan. Lokasi penyisipan adalah 4 *bit* LSB dari gambar itu sendiri.

MSB 2 bit	01	11	11	10	01	00
Semula	01110111	11111110	11000100	10110100	01010000	00001010
Bit yang disisipkan	0111	1110	0100	0111	1110	0100
Hasil	01110111	11111110	11000100	10110111	01011110	00000100

Gambar 7. Proses Embedding Langkah 3

Pada proses ekstraksi, jika diketahui piksel suatu gambar yang telah disisipi adalah seperti ditampilkan pada Gambar 8.

Komponen warna	R	G	B	R	G	B
warna yg tersisipi	01110111	11111110	11000100	10110111	01011110	00000100

Gambar 8. Proses Ekstraksi Langkah 1

Langkah berikutnya adalah membaca 4 *bit* LSB dari tiap-tiap komponen warna. Pada Gambar 9, terdapat keseluruhan 24 *bit*.

baca lsb	0111	1110	0100	0111	1110	0100
	Versi 1			Versi 2		

Gambar 9. Proses Ekstraksi Langkah 2

Karena pada proses penyisipan *bit* MSB terjadi proses penggandaan (Gambar 9), maka pada proses ekstraksi, kumpulan *bit* yang diperoleh dibagi ke dalam 2 bagian: versi 1 dan versi 2 (Gambar 10). Hal ini dimaksudkan agar jika terjadi perubahan pada gambar, maka dapat diketahui melalui setidaknya salah satu versi gambar yang disisipkan.

baca lsb	0111	1110	0100	0111	1110	0100
	Versi 1			Versi 2		
Versi 1	01000000	11000000	11000000	10000000	01000000	00000000
Versi 2	01000000	11000000	11000000	10000000	01000000	00000000

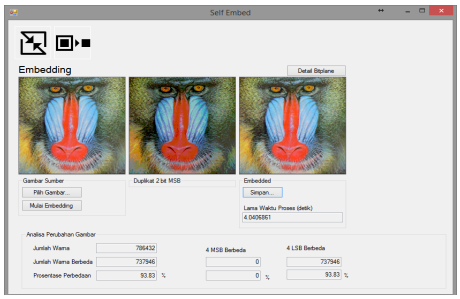
Gambar 10. Proses Ekstraksi Langkah 3

MSB dipilih karena merupakan *bit* yang paling mempengaruhi nilai warna pada suatu piksel. Semakin banyak *bit* MSB yang dipilih, semakin mirip dengan gambar keseluruhan. Oleh karena itu, MSB dipilih untuk disisipkan menjadi LSB. LSB merupakan *bit* yang jika mengalami perubahan nilai, tidak memberikan pengaruh yang signifikan, sehingga tepat digunakan untuk lokasi penyisipan data[7].

HASIL DAN PEMBAHASAN

Hasil Implementasi Sistem

Hasil implementasi sistem berdasarkan perancangan yang telah dilakukan, dijelaskan sebagai berikut. Gambar 11 menampilkan *form* yang digunakan untuk melakukan embeeding gambar. Pada *form* ini, disediakan fasilitas untuk memilih *file* gambar, melakukan proses *embedding* maupun ekstraksi, *image hasil* akan ditampilkan bersebelahan dengan*hostimage*, dan dapat melakukan pencatatan lama waktu proses, dan menyimpan hasil gambar. Pada *embedding*, proses yang terjadi adalah 2 *bit* MSB pada *hostimage* (gambar kiri) diambil menjadi duplikat 2 *bit* MSB (gambar tengah). Kemudian gambar duplikat ini disisipkan dua kali pada 4 *bit* LSB *hostimage*, sehingga menghasilkan embedded *image* (gambar kanan).

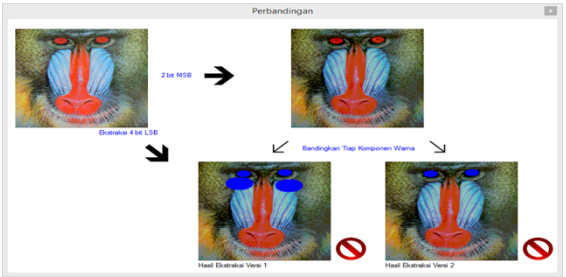


Gambar 11. Antarmuka untuk Proses Embedding

Gambar 12. Antarmuka untuk Proses Ekstraksi

Gambar 12 menampilkan *form* yang digunakan untuk melakukan ekstraksi gambar, dengan susunan dan aturan yang sama dengan *form embedding,* serta dapat menampilkan jumlah perbedaan warna. Pada ekstraksi, proses yang terjadi adalah 4 *bit* LSB pada *hostimage* dibaca (ekstrak), kemudian dibagi menjadi 2, masing-masing menjadi duplikat versi 1 dan duplikat versi 2. Tiap duplikat merupakan representasi dari 2 *bit* MSB citra digital asli.

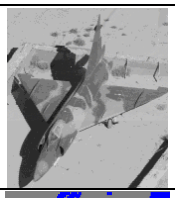
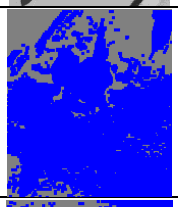
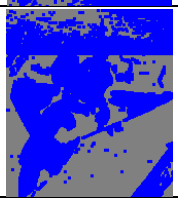
Pengujian visual untuk jenis perubahan dan hasil deteksi perubahan dilakukan untuk melihat apakah setelah melalui proses *embedding*,dan mengedit atau mengubah gambar sesuai jenis perubahan gambar (Manipulasi, rotate, resize, dan crop) mengalami perubahan pada warna. Pengujian dilakukan dengan melihat perbedaan warna antara citra digitial yang di ekstraksi 2bit MSB dan citra digital setelah terekstraksi 4bit LSB versi 1 dan versi 2. Hasil pengujian ditunjukkan pada Tabel 1-Tabel 4.



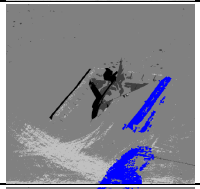
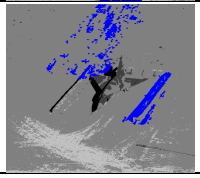
Gambar 13. Hasil Deteksi

Gambar 13 menunjukkan hasil ekstraksi 2bit MSB dan 4bit LSB versi 1 dan versi 2, serta menunjukkan lokasi pada citra digital, yang mengalami perubahan

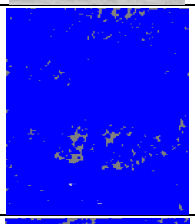
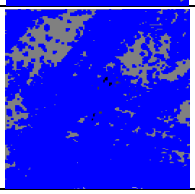
Tabel 1. Hasil Pengujian Visual untuk Jenis Perubahan Manipulasi

Jenis Perubahan	*Crop* *Gambar asli di-crop*
Hasil Perubahan	
Deteksi Versi 1	
Deteksi Versi 2	
Kesimpulan	Perubahan Gambar Terdeteksi.

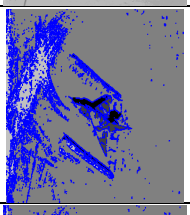
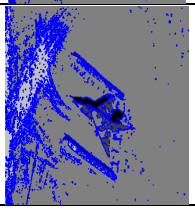
Tabel 2 Hasil Pengujian Visual untuk Jenis Perubahan *Resize*

Jenis Perubahan	Manipulasi Pagar sebelah kiri pesawat di hilangkan.
Hasil Perubahan	
Deteksi Versi 1	
Deteksi Versi 2	
Kesimpulan	Perubahan Gambar Terdeteksi.

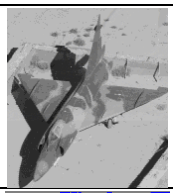
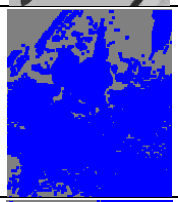
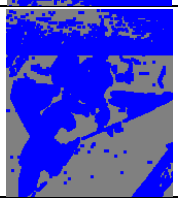
Tabel 3. Hasil Pengujian Visual untuk Jenis Perubahan *Rotate*

Jenis Perubahan	*Resize* Gambar asli diubah ukurannya dari 512 x 512 piksel menjadi 256 x 256 piksel.
Hasil Perubahan	
Deteksi Versi 1	
Deteksi Versi 2	
Kesimpulan	Perubahan Gambar Terdeteksi.

Tabel 4 Hasil Pengujian Visual untuk Jenis Perubahan *Crop*

Jenis Perubahan	Rotate Gambar asli di putar searah jarum jam 90 derajat.
Hasil Perubahan	
Deteksi Versi 1	
Deteksi Versi 2	
Kesimpulan	Perubahan Gambar Terdeteksi.

Tabel 5. Hasil Pengujian Visual untuk Jenis Perubahan *Crop*

Jenis Perubahan	*Crop* *Gambar asli di-crop*
Hasil Perubahan	
Deteksi Versi 1	
Deteksi Versi 2	
Kesimpulan	Perubahan Gambar Terdeteksi.

Berdasarkan hasil pengujian, disimpulkan bahwa kerusakan pada citra digital dapat terdeteksi apabila citra digital tersebut telah disisipi dengan digital watermark.

KESIMPULAN

Berdasarkan penelitian, pengujian dan analisis terhadap sistem, maka dapat diambil kesimpulan sebagai berikut: (1) *Self embedding* dapat dimanfaatkan sebagai *digital watermarking* sekaligus untuk deteksi perubahan/kerusakan pada citra digital; (2) Proses *embedding* gambar dapat dilakukan dengan menyisipkan 2*bit* MSB sebanyak 2 kali yang menghasilkan gambar versi 1 dan versi 2 yang berfungsi untuk melakukan pengecekan perubahan yang terjadi pada citra digital; (3) Berdasarkan hasil pengujian visual untuk jenis perubahan, dan hasil deteksi perubahan, diketahui bahwa setiap gambar hasil *embedding* yang dilakukan beberapa jenis perubahan (manipulasi, *rotate*, *resize*, dan *crop*), menunjukkan perbedaan warna yang berbeda pada versi 1 dan versi 2, sehingga dapat diketahui jika suatu gambar yang telah melalui proses *embedding* sudah. Perubahan warna yang terjadi di tunjukan dengan warna biru,sehingga apabila suatu gambar yang telah melaui proses *embedding* mengalami perubahan dapat segera diketahui letak perubahan gambar tersebut; Saran pengembangan yang dapat diberikan untuk penelitian lebih lanjut adalah sebagai berikut: (1) Duplikat yang disisipkan dapat disandikan terlebih dahulu dengan suatu algoritma kriptografi, sehingga menambah tingkat keamanan; (2) Duplikat yang disisipkan dapat dibuat lebih dari 2 versi, dengan tujuan untuk menunjukkan lokasi perubahan yang lebih detil.

DAFTAR PUSTAKA

[1]. Singh, P. & Chadha, R. S. 2013. A Survey of Digital Watermarking Techniques , Applications and Attacks. 2, 165–175.

[2]. Cummins, J., Diskin, P., Lau, S. & Parlett, R. 2004. Steganography And Digital Watermarking.

[3]. Lusson, F., Bailey, K., Leeney, M. & Curran, K. 2013. A novel approach to digital watermarking, exploiting colour spaces. Signal Processing 93, 1268–1294. (doi:10.1016/j.sigpro.2012.10.018)

[4]. Yeung, M. M. & Mintzer, F. 1997. An invisible watermarking technique for image verification. Proceedings of International Conference on Image Processing , 680–683. (doi:10.1109/ICIP.1997.638587)

[5]. Qian, Z., Feng, G., Zhang, X. & Wang, S. 2010. Image self-embedding with high-quality restoration capability. (doi:10.1016/j.dsp.2010.04.006)

[6]. Johnson, N. F., Duric, Z. & Jajodia, S. 2001. Information hiding: steganography and watermarking: attacks and countermeasures. Springer.

[7]. Podesser, M., Schmidt, H.-P. & Uhl, A. 2012. Selective bitplane encryption for secure transmission of image data in mobile environments. Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02). , 4–6.

[8]. Cheddad, A., Condell, J., Curran, K. & Kevitt, P. M. 2009. A secure and improved self-embedding algorithm to combat digital document forgery. Signal Processing

[9]. Wu, M. 1998. Watermarking for image authentication. Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269) 2, 437–441. (doi:10.1109/ICIP.1998.723413)

[10].Parvez, M. T. & Gutub, A. A.-A. 2008. RGB Intensity Based Variable-Bits Image Steganography. 2008 IEEE Asia-Pacific Services Computing Conference , 1322–1327. (doi:10.1109/APSCC.2008.105)