TEKNIK PENYISIPAN PADA DATA DIGITAL DENGAN MENGGUNAKAN METODE LEAST SIGNIFICANT BIT

Sapta Aji Sri Margiutomo, Retno Sundari[*)]

ABSTRACT

*Changes in the presentation of data from analog to digital turned out to bring the issue of copyright protection and authentication for digital image. This is due to the ease with which duplicates can be made to the digital image.Digital watermaking can be applied to insert a copyright label into a digital image of the invisible. This will be difficult for an unauthorized person to remove the copyright label so that only authorized parties are able to reveal the label copyright. The method used is the method of LSB (*Least *Significant Bit*).

Keywords : Digital Watermaking, LSB

Citra digital merupakan suatu fungsi dengan nilai-nilai yang berupa **intensitas cahaya pada tiap-tiap titik** pada bidang yang telah diquantisasikan (diambil sampelnya pada interval diskrit). Titik dimana suatu gambar di-sampling disebut *picture element* (pixel). Nilai intensitas warna pada suatu pixel disebut *gray scale level* .

1 bit→ *binary-valued image* (0-1)

8 bits→ *gray level* (0-255)

16 bits→ *high color* (2^{16})

24 bits→ 2^{24} *true color*

32 bits→ *true color* 2^{32}

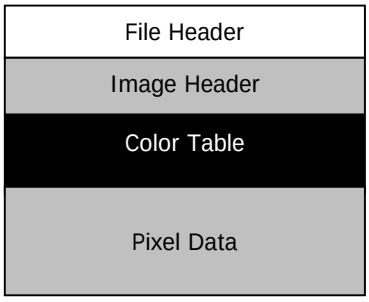
Sedangkan format *fie* BMP menurut Syahbana (2007:5) :

Format *file* BMP (Bitmap Image) merupakan format grafis yang fleksibel untuk platform Windows sehingga dapat dibaca oleh program grafis manapun. Format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit sampai 24 bit. Format *file* ini mampu menyimpan gambar dalam mode warna RGB, *Grayscale*, *Indexed Color*, dan Bitmap.

Masih mengenai format *file* BMP, Rachmat & Roswanto (2005:11) mengemukakan bahwa "format *file* BMP adalah format yang belum terkompresi dan menggunakan sistem warna RGB (*Red, Green, Blue*) dimana masing-masing warna pixelnya terdiri dari 3 komponen tersebut yang dicampur menjadi satu"

Citra digital BMP menurut Rachmat & Roswanto (2005:6-8) adalah sebagai berikut:

Citra digital BMP memiliki informasi tambahan seperti lebar x panjang gambar, kedalaman gambar, pembuat, dll. Informasi-informasi tersebut disimpan pada bagian tertentu dari struktur file. Beberapa bagian dari struktur *file* gambar antara lain *file header*, *image header*, *color table* dan pixel data.



Gambar 1. Struktur *File* Gambar BMP

Watermarking

Berikut ini sejarah *watermarking* yang diulas Munir (2006:310) bahwasanya:

*) Dosen STMIK PPKIA Pradnya Paramita Malang

Watermarking sudah ada sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi *watermark* atau tanda-air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan terbentuklah suatu kertas ber-*watermark*. Kertas ini biasanya digunakan oleh seniman atau sastrawan untuk menulis karya mereka. Kertas yang sudah dibubuhi tanda-air tersebut sekaligus dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka.

Ide *watermarking* pada data digital (sehingga disebut *digital watermarking*) dikembangkan di Jepang tahun 1990 dan di Swiss tahun 1993. *Digital watermarking* semakin berkembang seiring dengan semakin meluasnya penggunaan internet, objek digital seperti video, citra, dan suara yang dapat dengan mudah digandakan dan disebarluaskan.

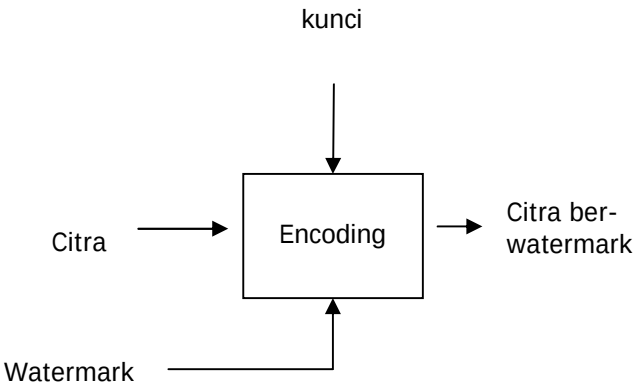
Sekilas Mengenai *Watermarking* Digital

Menurut Munir(2006:311-313) bahwasanya:

Teknik *watermarking* secara umum terdiri dari 2 tahapan yakni:

- Penyisipan *watermark* (*watermark embedding*)
- Ekstraksi atau pendeteksian *watermark*

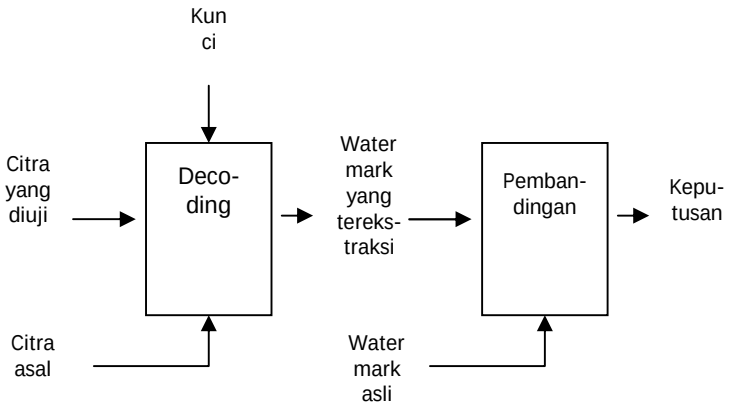
Penyisipan *watermark* dapat dipandang sebagai superposisi data *watermark* pada citra dengan suatu cara sedemikian sehingga superposisi tersebut tidak mempengaruhi persepsi visual terhadap citra. Gambar 2.6 memperlihatkan sebuah *encoder* yang melakukan penyisipan *watermark*. *Encoder* menerima masukan berupa citra, *watermark*, dan kunci. *Encoder* menghasilkan citra ber-*watermark*. Citra asal dan citra ber-*watermark* hampir mirip secara statistik, atau secara visual mempunyai persepsi yang sama.



Gambar 2. Proses Penyisipan *Watermark* ke dalam Citra

Watermark harus dapat diekstraksi atau dideteksi kembali. *Decoder* menerima masukan berupa citra uji, kunci *k*, dan menghasilkan *watermark* terekstrak. *Decoder* dapat mengikutsertakan citra asal yang belum diberi *watermark* (*non-blind watermarking*) atau tidak sama sekali (*blind watermarking*). Selanjutnya *watermark* terekstraksi dibandingkan dengan *watermark* asli dengan fungsi komparator untuk menghasilkan keputusan berupa keluaran biner (1 menyatakan cocok, 0 menyatakan sebaliknya).

*) Dosen STMIK PPKIA Pradnya Paramita Malang



Gambar 3. Proses Ekstraksi dan Verifikasi *Watermark* dari dalam Citra

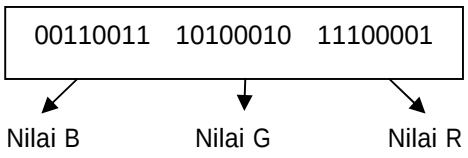
Sebuah teknik *watermarking* yang bagus harus memenuhi persyaratan berikut:

1. *Imperceptibility* : keberadaan *watermark* tidak dapat dipersepsi oleh indra visual. Hal ini bertujuan untuk menghindari gangguan pengamatan visual.
2. *Key uniqueness*: kunci yang berbeda seharusnya menghasilkan *watermark* yang berbeda. Ini berarti penggunaan kunci yang salah dapat menyebabkan hasil ekstraksi/deteksi *watermark* yang salah pula.
3. *Noninvertibility*: secara komputasi sangat sukar menemukan *watermark* bila diketahui hanya citra ber-*watermark* saja.
4. *Image dependency*: satu kunci menghasilkan sebuah *watermark* tunggal, tetapi *watermark* bergantung pada isi citra. Salah satu pendekatan yang digunakan adalah membangkitkan *watermark* dari nilai *hash* (*message digest*) citra asli, sebab nilai *hash* bergantung pada isi citra.
5. *Robustness*: *watermark* seharusnya tetap kokoh terhadap berbagai serangan yang dilakukan pada citra ber-*watermark*. Ini berarti manipulasi yag dilakukan terhadap citra ber-*watermark* tidak merusak *watermark* (*watermark* masih dapat dideteksi).

Metode LSB untuk *Image Watermarking*

Menurut Munir (2006:308) bahwasanya: LSB merupakan metode *image watermarking* yang sederhana dan mudah diimplementasikan. Setiap *pixel* dalam citra BMP berukuran 1 sampai 3 *byte*. Diasumsikan *pixel* citra yang digunakan memiliki mode warna RGB berukuran 24 bit (8x3 *byte*). Masing-masing *channel* RGB memiliki MSB (*Most Significant Bit*) yakni bit pertama dari 8 bit maupun LSB (*Least Significant Bit*) yakni bit terakhir dari 8 bit. Bit yang cocok digunakan untuk menyisipkan *watermark* adalah LSB karena perubahan 1 bit ke atas ataupun ke bawah akan sulit dipersepsi oleh indra visual manusia.

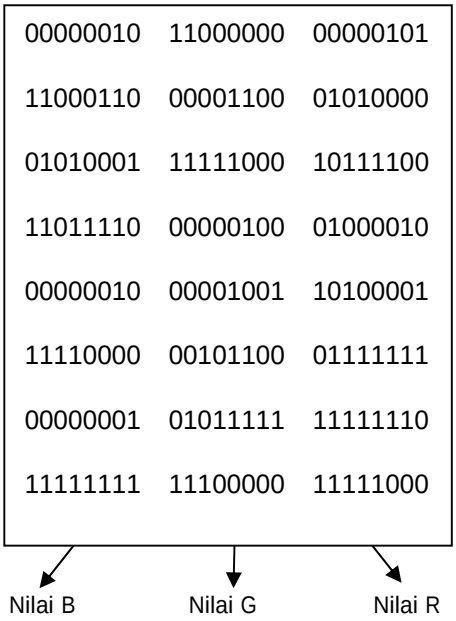
Sebagai ilustrasi, misalkan suatu *pixel* citra *watermark* memiliki nilai biner seperti Gambar 4 di bawah ini.



Gambar 4. Satu *Pixel* Citra *Watermark*

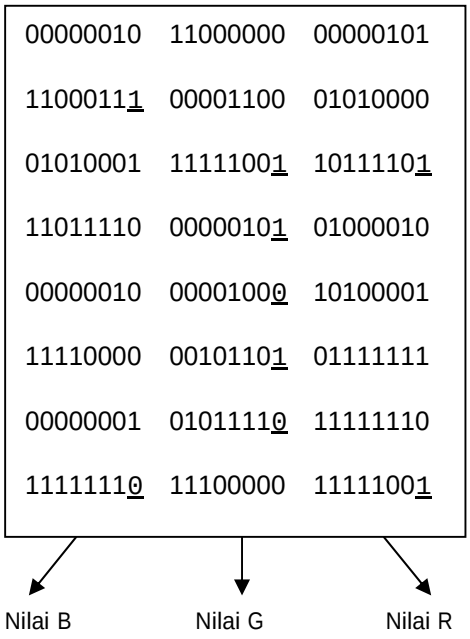
Karena dalam metode ini 1 *byte* citra media hanya mampu menyimpan 1 bit *watermark* maka diperlukan 8 *pixel* citra media. Dimisalkan sebuah citra media memiliki barisan biner sebagai berikut, dimana tiap bit dari *channel watermark* akan disisipkan ke bit-bit terakhir *channel* citra media:

*) Dosen STMIK PPKIA Pradnya Paramita Malang



Gambar 5. 8 *Pixel* Citra Media

Maka setelah disisipi bit-bit *watermark*, bit-bit citra media 8 *pixel* pada Gambar 5 akan menjadi seperti pada Gambar 6 di bawah ini.



Gambar 6. 8 *Pixel* Citra Media Ber-*Watermark*

Dari gambar 6 dapat dilihat bahwasanya setelah penyisipan *watermark*, bit-bit yang berubah hanya 9 bit LSB dari 24 *byte* ditunjukkan dengan tanda angka bergaris bawah bercetak tebal.

Dengan banyaknya *pixel* media yang diperlukan untuk menampung 1 pixel *watermark* yakni 8 *pixel* citra media, maka dalam penelitian ini dilakukan modifikasi metode LSB. Berdasarkan penelitian Swanson, Zhu, Chau & Tewfik (1997) bahwa "sistem penglihatan manusia (*human visual system*) tidak terlalu peka pada komponen warna biru" maka dalam penelitian ini penyisipan dilakukan sebanyak 4 bit pada *channel* warna biru dan masing-masing 1 bit pada *channel* hijau dan merah. Maka untuk 1 *pixel* citra *watermark* diperlukan 4 *pixel* citra media.

Metode MD5 untuk Unikasi *Watermark*

Dalam bukunya, Munir (2006:220) menulis bahwa "MD5 adalah fungsi *hash* satu arah yang dibuat oleh Ronald Rivest pada tahun 1991. Algoritma MD5 menerima masukan berupa pesan dengan ukuran sembarang dan menghasilkan *message digest* yang panjangnya 128 bit."

Sebagaimana diulas oleh Ramayanti (2007:9) bahwasannya:

Message digest atau intisari pesan harus mempunyai tiga sifat penting, yaitu:

1. Bila P diketahui, maka MD(P) akan dengan mudah dapat dihitung.
2. Bila MD(P) diketahui, maka tidak mungkin menghitung P.
3. Tidak seorang pun dapat memberi dua pesan yang mempunyai intisari pesan yang sama. H(M) ≠ H(M).

*) Dosen STMIK PPKIA Pradnya Paramita Malang

Menurut Munir (2006:220-224) bahwasanya: Langkah-langkah pembuatan *message digest* dengan MD5 adalah sebagai berikut:

1. Penambahan bit-bit pengganjal (*padding bits*).

   Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512. Ini berarti panjang pesan setelah ditambahi bit-bit pengganjal adalah 64 bit kurang dari kelipatan 512.
2. Penambahan nilai panjang pesan semula.

   Pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Jika panjang pesan > 2^{64} maka yang diambil adalah panjangnya dalam modulo 2^{64}. Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512 bit.
3. Inisialisasi penyangga (*buffer*) MD.

   MD5 membutuhkan 4 buah penyangga yang masing-masing panjangnya 32 bit. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

   A = 67452301

   B = EFCDAB89

   C = 98BADCFE

   D = 10325476
4. Pengolahan pesan dalam blok berukuran 512 bit.Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y_0 sampai Y_{L-1}). Setiap blok 512 bit diproses bersama dengan penyangga MD menjadi keluaran 128 bit, dan ini disebut proses H_{MD5}. Proses H_{MD5} terdiri dari 4 putaran, dan masing-masing putaran melakukan 16 kali operasi dasar MD5 dan tiap operasi dasar memakai sebuah elemen T yang didapatkan dari perhitungan fungsi $T(i)=2^{32}$ x abs(sin(i)) dimana i dalam radian. Berikut adalah tabel fungsi untuk tiap putaran MD5.

Tabel 1. Fungsi-Fungsi Dasar MD5

Nama	**Notasi**	**g(b,c,d)**
f_F	F(b,c,d)	$(b \wedge c) \vee (\sim b \wedge d)$
f_G	G(b,c,d)	$(b \wedge d) \vee (c \wedge \sim d)$
f_H	H(b,c,d)	$b \oplus c \oplus d$
f_I	I(b,c,d)	$c \oplus (b \vee \sim d)$

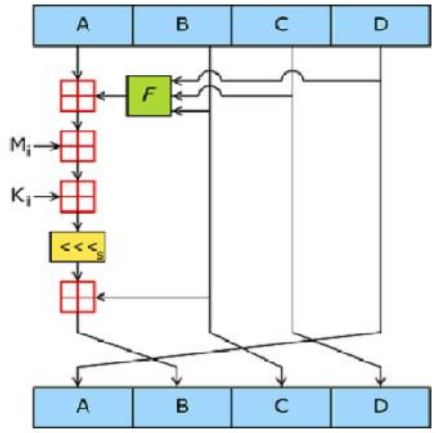
Catatan: operator logika AND, OR, NOT, XOR masing-masing dilambangkan dengan $\wedge$, $\vee$, $\sim$, $\oplus$

Pada Gambar 7 sebagaimana diulas oleh Ramayanti (2007:11-12) :

Dapat dilihat satu buah operasi dari MD-5 dengan operasi yang dipakai sebagai contoh adalah FF(a,b,c,d,Mj,s,T(i)) menunjukkan:

a = b+((a+F(b,c,d)+Mj +T(i0) <<< s)

*) Dosen STMIK PPKIA Pradnya Paramita Malang



Gambar 7. Satu buah operasi MD5

Bila Mj menggambarkan pesan ke-j dari sub blok (dari 0 sampai 15) dan <<< s menggambarkan bit akan digeser ke kiri sebanyak s bit, maka keempat operasi dari masing-masing ronde adalah:

FF(a,b,c,d,Mj,s,T(i)) menunjukan a = b+((a+F(b,c,d)+Mj+ T(i)) <<< s)

FG(a,b,c,d,Mj,s, T(i)) menunjukan a = b+((a+G(b,c,d)+Mj+ T(i)) <<< s)

FH(a,b,c,d,Mj,s, T(i)) menunjukan a = b+((a+H(b,c,d)+Mj+ T(i)) <<< s)

FI(a,b,c,d,Mj,s, T(i)) menunjukan a = b+((a+I(b,c,d)+Mj + T(i)) <<< s)

5. Keluaran MD5

   Keluaran dari MD5 adalah 128 bit dari *word* terendah A dan tertinggi *word* D masing-masing 32 bit.

Watermarking pada citra digital menggunakan metode LSB berfungsi untuk menyisipkan label pada citra media BMP. Bit citra media BMP yang digunakan untuk penyisipan yakni 1 bit pada *channel* warna merah dan hijau, sedangkan untuk *channel* biru adalah sebanyak 4 bit.

Label yang akan disisipkan adalah berupa gambar BMP. Hal ini untuk menyiasati kelemahan metode LSB yang mudah dirusak dengan mengganti bit terakhir. Dengan label berupa gambar, meski dilakukan penggantian pada bit terakhir, saat proses pendeteksian label masih dapat menghasilkan gambar label yang mendekati gambar aslinya, karena kelemahan mata manusia yang tidak dapat membedakan selisih warna yang berdekatan sehingga label hak cipta masih terbaca.

Metode MD5 digunakan untuk unikasi label *watermark*. Hal ini dimaksudkan karena *watermark* yang baik adalah unik untuk tiap citra media. Hasil proses MD5 disisipkan ke citra label untuk kemudian bersama-sama disisipkan ke citra media.

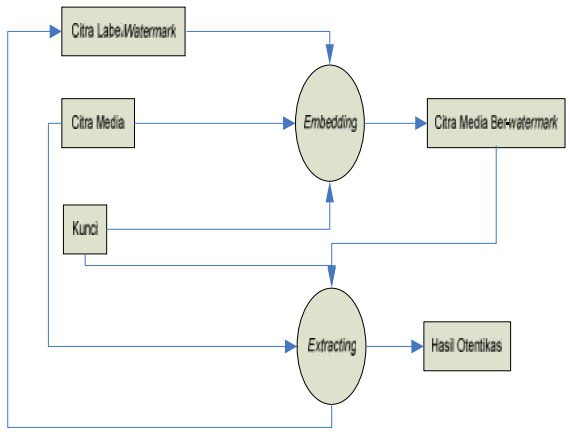
Untuk keperluan pengungkapan kembali *watermark*, diperlukan penyimpanan panjang dan lebar *watermark* pada *header* citra media. Namun akan sangat riskan jika panjang dan lebar *watermark* disimpan dalam format bit asli. Oleh karenanya dilakukan enkripsi XOR terhadap panjang dan lebar label kemudian hasil enkripsi disimpan pada *header* citra media.

Metode Penelitian

Penelitian ini membuat aplikasi yang dibangun adalah aplikasi *watermarking*, yakni aplikasi yang digunakan untuk menyisipkan label hak cipta pada citra media BMP dan mengungkap kembali label yang sudah disisipkan untuk mengetahui hak cipta suatu citra BMP serta mendeteksi otentikasi citra BMP tersebut.

*) Dosen STMIK PPKIA Pradnya Paramita Malang

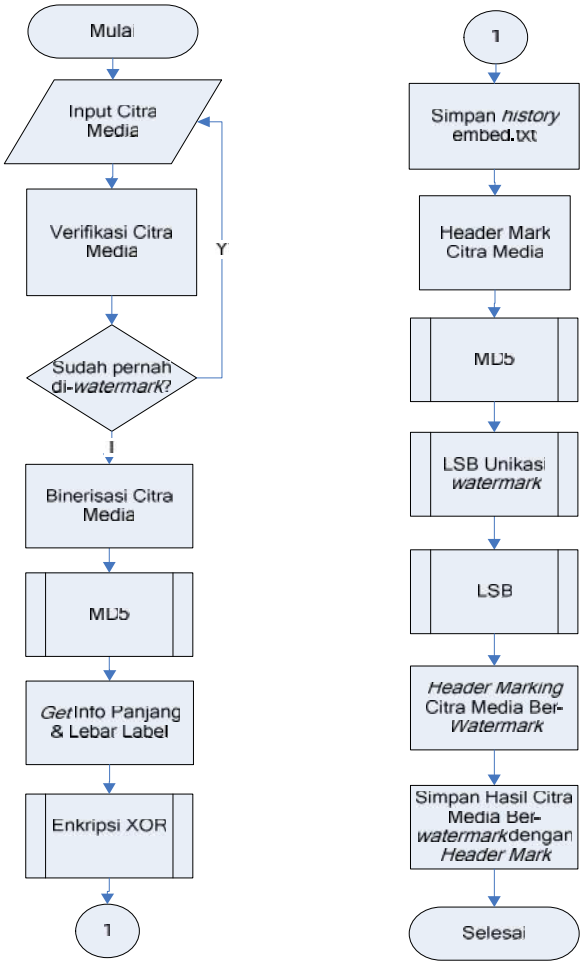
Secara blok diagram, aplikasi ini dapat digambarkan sebagai berikut:



Gambar 8. Blok Diagram Aplikasi *Watermarking*

Proses Penyisipan (*Embedding*) Label

Proses penyisipan (*embedding*) label pada aplikasi ini meliputi proses verifikasi citra media untuk mengetahui citra sudah pernah mengalamai *digital watermarking* atau tidak, proses MD5 citra asli, pembacaan data panjang dan lebar citra label, enkripsi XOR panjang dan lebar citra label dengan *key* yakni kunci *watermarking*, penyimpanan kode citra media serta panjang dan lebar label ke embed.txt, penyisipan nomor media sebagai objek MyImage pada *header* media, MD5 media dengan *header* nomor, penyisipan hasil MD5 ke dalam citra label (LSB unikasi *watermark*), proses penyisipan label ber-MD5 ke dalam citra media (LSB), penyimpanan *cipher* panjang dan lebar citra label ke dalam citra media (*header marking*), dan yang terakhir adalah penyimpanan citra hasil *watermarking*. Berikut ini *flowchart* untuk memperjelas proses penyisipan (*embedding*) label:



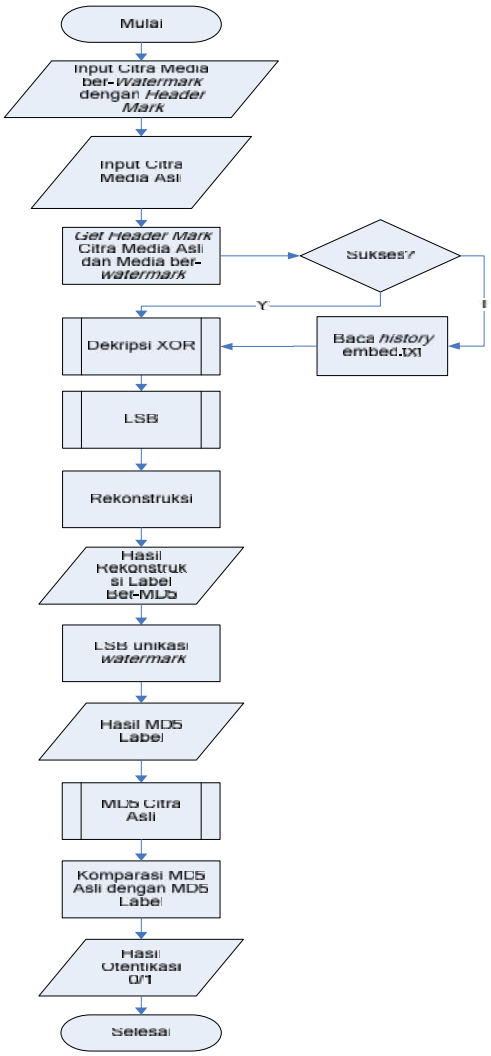
Gambar 9.
Flowchart* Proses *Embedding

Proses Pengungkapan (*Extracting*) Label

Proses pengungkapan (*extracting*) label pada aplikasi ini dijelaskan pada Gambar 3.3 yakni meliputi proses pembacaan *header* citra media ber-*watermark* untuk mendapatkan panjang dan lebar label, pembacaan *history* embed.txt jika proses pembacaan *header* gagal, dekripsi XOR panjang dan lebar label, pembacaan bit label dari citra media ber-*watermark* (LSB), rekonstruksi label berdasar panjang dan lebarnya, deteksi bit MD5 pada label (LSB unikasi *watermark*), proses MD5 citra asli, komparasi MD5 citra asli dengan MD5

*) Dosen STMIK PPKIA Pradnya Paramita Malang

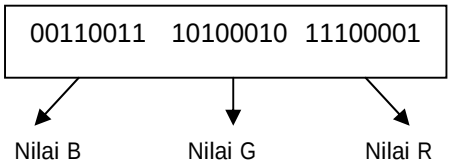
dari label untuk mendapatkan otentikasi citra ber-*watermark*.



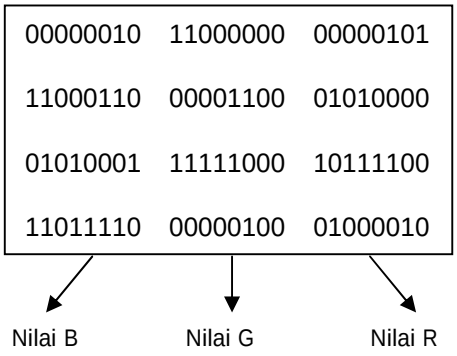
Gambar 10. *Flowchart* Proses *Extracting*

Metode LSB pada Aplikasi *Watermarking*

Aplikasi ini mengimplementasi metode LSB untuk proses unikasi *watermark* dan proses *watermarking* itu sendiri. Hanya saja, ada modifikasi terhadap metode LSB yakni jumlah bit label yang disisipkan tidak sama untuk masing-masing *channel* media. Berdasarkan hasil penelitian pakar yang menyimpulkan bahwasanya sistem penglihatan manusia (*human visual system*) tidak terlalu peka pada komponen warna biru, maka pada aplikasi ini dilakukan modifikasi penyisipan label yakni 1 bit untuk *channel* merah dan hijau, sedangkan untuk *channel* biru disisipkan sebanyak 4 bit. Diharapkan dengan pola 4-1-1 untuk urutan *channel* biru-hijau-merah, pixel citra media yang diperlukan untuk menyisipkan 24 bit citra label hanyalah 4 pixel. Contohnya adalah sebagai berikut, misalkan suatu *pixel* citra *watermark* memiliki nilai biner seperti Gambar 11. Dan 4 *pixel* citra media untuk nilai biner seperti Gambar 12. Dan 4 *pixel* citra media untuk disisipi citra label/*watermark* adalah pada Gambar 11 sebagai berikut:



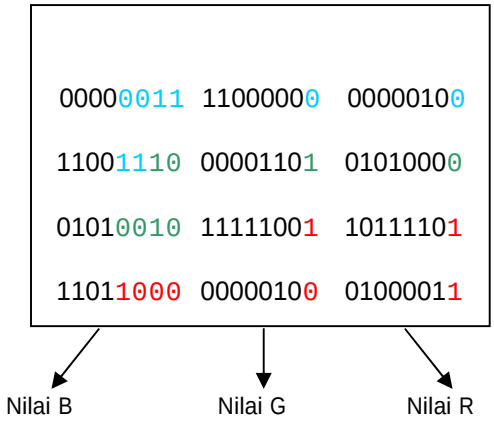
Gambar 11. Satu *Pixel* Citra *Watermark*



Gambar 12. 4 *Pixel* Citra Media

Dengan modifikasi LSB yakni menggunakan pola 4-1-1 maka hasil penyisipan adalah seperti pada Gambar 13 berikut:

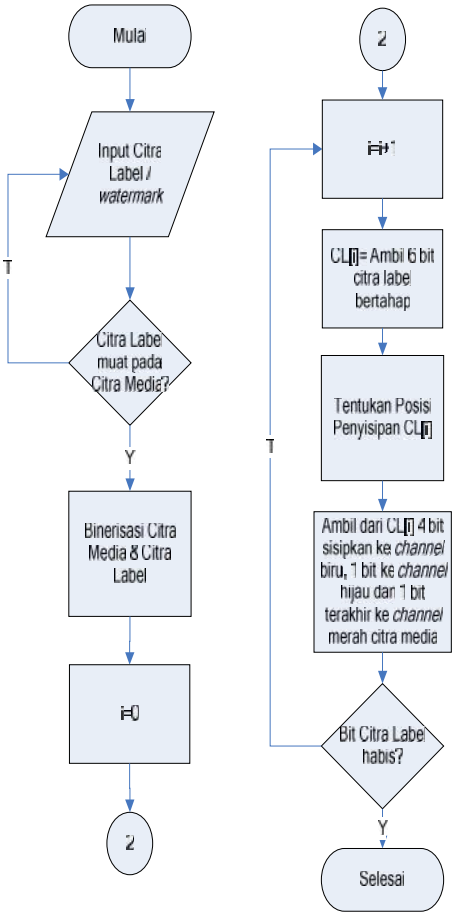
*) Dosen STMIK PPKIA Pradnya Paramita Malang



Gambar 13. 4 *Pixel* Citra Media Berlabel

Warna merah, hijau dan biru digunakan untuk memudahkan pembacaan bit-bit label disesuaikan dengan *channel*.

Sedangkan untuk LSB unikasi *watermark*, pada dasarnya sama dengan LSB di atas. Perbedaannya hanyalah data yang disisipkan bukan dalam bentuk gambar melainkan 128 bit MD5 citra media. Sehingga proses binerisasi hanya mengolah data citra media karena bit MD5 sudah dalam format biner. Berikut ini *flowchart* untuk memperjelas algoritma LSB:

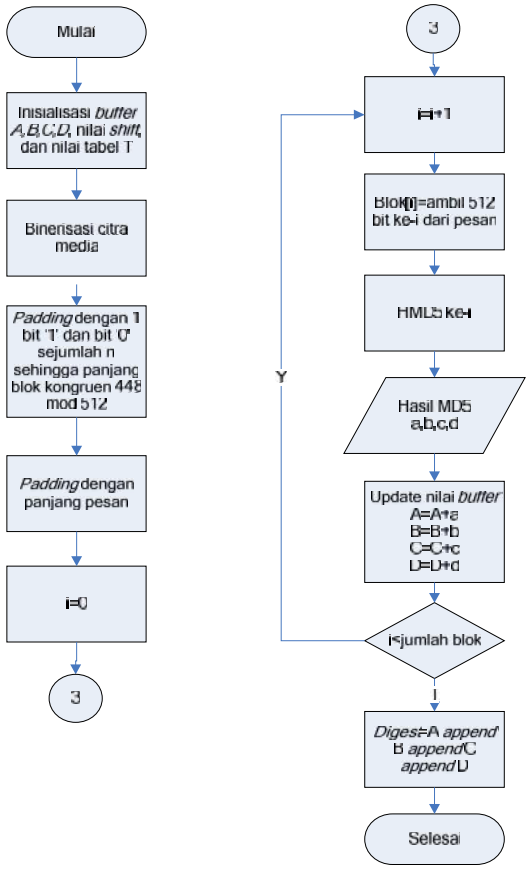


Gambar 14. *Flowchart* Algoritma LSB

Metode MD5 pada Aplikasi *Watermarking*

Metode MD5 digunakan untuk unikasi *watermark*, dimana tiap citra media memiliki nilai *message digest* unik yang disisipkan ke label dengan metode LSB sehingga label yang didapatkan adalah unik sesuai dengan citra media. Berikut ini adalah *flowchart* algoritma MD5:

*) Dosen STMIK PPKIA Pradnya Paramita Malang



Gambar 15. *Flowchart* MD5

Penyisipan MD5 citra media ke label juga dimaksudkan untuk otentikasi citra media berlabel yang dipublikasikan. Otentikasi ini memanfaatkan sifat MD5 yang akan berubah jika ada modifikasi pada citra media walaupun perubahannya hanya 1 bit. Sehingga pada proses komparasi MD5 citra media asli dengan citra media berlabel didapatkan nilai otentikasi yakni 1 yang berarti citra otentik dan 0 yang berarti citra tidak otentik.

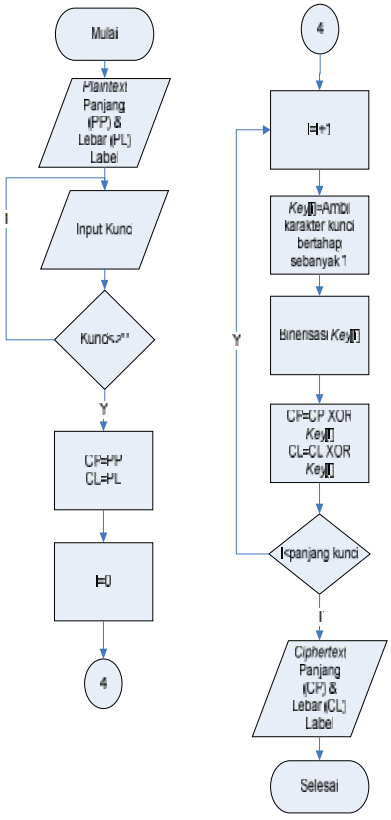
Enkripsi XOR pada Aplikasi *Watermarking*

XOR digunakan untuk enkripsi *plaintext* panjang dan lebar label yang dibutuhkan pada saat rekonstruksi label. Panjang dan lebar label di-XOR dengan kunci yang dimasukkan oleh *user*. Proses enkripsi dilakukan berulang dengan menggunakan 1 karakter kunci secara bertahap. Hal ini dilakukan untuk mempersulit kriptoanalisis *cipher*.

Ciphertext panjang dan lebar hasil enkripsi disimpan pada *header* citra media (*header marking*) untuk diambil kembali pada saat *extracting* label, kemudian didekripsi ke bentuk awal. Hasil *plaintext* panjang dan lebar ini kemudian digunakan untuk rekonstruksi label.

Proses dekripsi adalah kebalikan dari proses enkripsi. Alur prosesnya adalah sama, juga dilakukan secara berulang dan bertahap seperti pada proses enkripsi. Namun sebagai *input* proses adalah *cipher* panjang dan lebar yang diambil dari *header* citra hasil *watermarking*.

Proses enkripsi secara detail dijelaskan pada Gambar 16 di bawah ini :

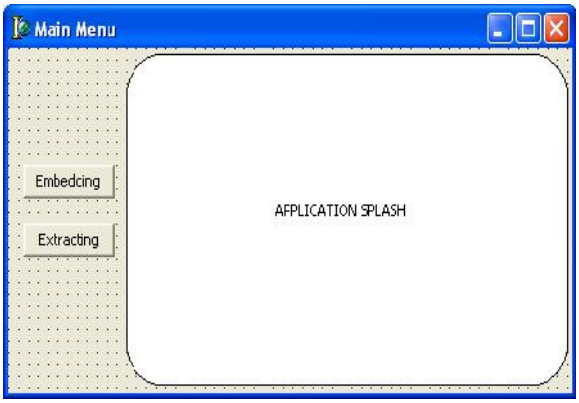


Gambar 16. *Flowchart* Enkripsi XOR

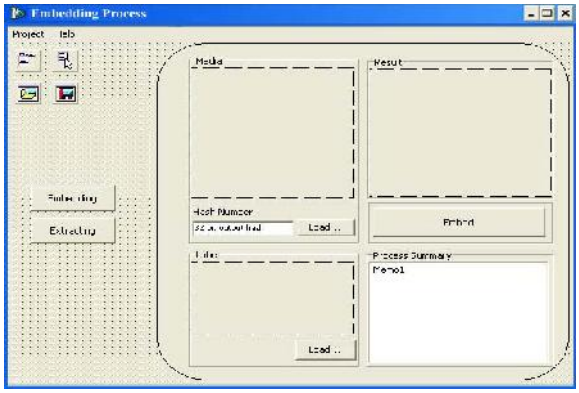
*) Dosen STMIK PPKIA Pradnya Paramita Malang

Tampilan Aplikasi *Watermarking*

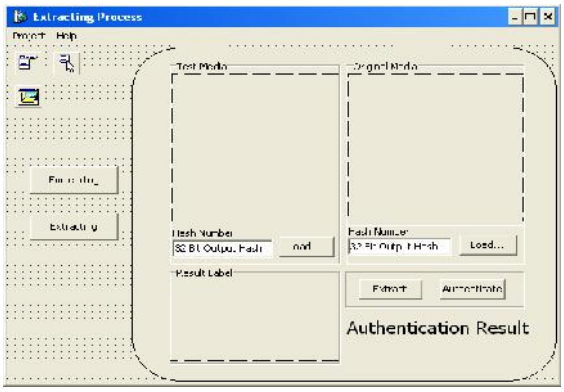
Dalam sub bab ini diberikan rancangan tampilan aplikasi *watermarking*, antara lain menu utama, *form* proses *embedding*, *form* proses *extracting* dan *form* informasi tentang aplikasi ini.



Gambar 17.
Tampilan Menu Utama Aplikasi *Watermarking*



Gambar 18.
Tampilan Proses *Embedding* Aplikasi *Watermarking*



Gambar 19.
Tampilan Proses *Extracting* Aplikasi *Watermarking*

Gambar 20. Tampilan Info Aplikasi *Watermarking*

Definisi Istilah

Berikut ini definisi beberapa istilah yang digunakan dalam penulisan untuk mempermudah pemahaman pembaca:

1. Spasial adalah acuan pada posisi, obyek, dan hubungan di antaranya dalam ruang bumi.
2. Nilai *Hash* adalah hasil dari pemetaan dari domain yang lebih besar (*message* sebagai *input*) ke *range* yang lebih kecil (nilai *hash* sebagai *output*).
3. *Cropping* adalah proses memotong sebagian gambar seperti bagian yang tidak dibutuhkan dari sebuah grafik atau kelebihan ruang putih disekitar bingkai.
4. Kompresi adalah proses pemampatan ukuran data dengan suatu cara tertentu, sehingga menghemat media yang digunakan.
5. *Encoder* adalah alat yang digunakan untuk mengacak suatu informasi.
6. *Decoder* adalah alat yang digunakan untuk mengembalikan suatu informasi yang telah diacak.
7. *Plaintext* adalah teks yang bisa dibaca oleh manusia dan dalam bentuk belum dienkripsi.

*) Dosen STMIK PPKIA Pradnya Paramita Malang

8. *Ciphertext* adalah text informasi yang telah diubah penulisannya menjadi kode rahasia, sehingga tidak ada orang yang mampu menggunakannya kecuali dia mempunyai kata kunci untuk membukanya.
9. Radian adalah salah satu satuan ukuran selain derajat. Satuan ini didefinisikan sebagai panjang garis lengkung pada suatu lingkaran.

Kesimpulan

Berdasarkan analisa yang dilakukan, dengan adanya implementasi *digital watermarking* yang sudah dibuat serta pengujian yang dilakukan dapat diambil kesimpulan bahwasanya:

1. Dilihat dari kesamaan visual antara citra asli dan citra hasil proses *embedding*, dapat disimpulkan bahwasanya metode LSB (*Least Significant Bit*) cukup handal digunakan sebagai metode untuk *digital watermarking* dengan bahasa pemrograman Delphi 6, dikarenakan tingkat invisibilitasnya yang tinggi.
2. Dengan prosentase keberhasilan otentikasi sebesar 80%, maka dapat disimpulkan bahwasanya metode MD5 (*Message Digest* 5) cukup handal digunakan untuk otentikasi citra dengan bahasa pemrograman Delphi 6.
3. Rata-rata keberhasilan pembacaan label hak cipta hasil *digital watermarking* dengan mengimplementasi metode LSB (*Least Significant Bit*) dan MD5 (*Message Digest* 5), setelah diberi serangan berupa *cropping*, kompresi GIF, perubahan kontras, penggandaan dan pemberian *noise* adalah 55,55% berarti cukup *robust.*

Saran

Dari implementasi *digital watermarking* yang sudah dibuat serta pengujian yang sudah dilakukan, terdapat beberapa kelebihan dan kekurangan. Kelebihannya antara lain kesederhanaan implementasi metode LSB yang tidak menuntut model matematika yang rumit serta kemampuan invisibilitas hasil metode LSB yang tinggi, dilihat dari kesamaan visual antara citra asli dan citra yang sudah diberi label. Sedangkan kekurangannya antara lain bahwasanya bit-bit hasil metode LSB kurang mampu bertahan dari proses pengolahan citra yang bisa merubah nilai bit. Kekurangan lainnya adalah nilai otentikasi kurang *valid* jika terjadi modifikasi kecil yang tidak mengenai bit tempat menyimpan nilai MD5 pada citra hasil *embedding.*

DAFTAR RUJUKAN

http://id.wikipedia.org/wiki/Enkripsi (diakses tanggal 15 November 2007).

Munir, Rinaldi. 2005. *Image Watermarking untuk Memproteksi Citra Digital dan Aplikasinya pada Citra Medis*. Bandung: Institut Teknologi Bandung.

Munir, Rinaldi. 2006. *Kriptografi*. Bandung: Informatika.

Puspitasari, Ratna. 2007. *Aplikasi Watermarking Guna Pelabelan Citra Digital Menggunakan Metode LSB*. Malang: Join Program BA.

*) Dosen STMIK PPKIA Pradnya Paramita Malang

Zakaria, Teddy Marcus. 2003. *Pemrograman Delphi untuk Pemula: IDE dan Struktur Pemrograman*.

http://www.ilmukomputer.com Zine.

http://ezine.echo.or.id/ezine14/04_Algoritma_Enkripsi_One_Time_Pad.txt (*download* tanggal 15 November 2007).

W. Bender, D. Gruhl, N. Morimoto, A. Lu, *Techniques for data hiding*, IBM System Journal, Vol. 35, 1996.

Lintian Qiao, *Multimedia Security and Copyright Protection*, 1998.

J. D. Gordy & L. T. Bruton, *Performance Evaluation of Digital Audio Watermarking Algorithms*.

Suhono H. Supangkat, Juanda, Riwut Libinuko, *Proteksi Citra Digital dengan Informasi yang tidak kelihatan*, ITB, 2000

*) Dosen STMIK PPKIA Pradnya Paramita Malang