

WATERMARKING PADA CITRA DIGITAL MENGGUNAKAN KOMBINASI PIXEL VALUE INDICATOR DAN METODE MOST SIGNIFICANT BIT

Posma Lumbanraja

Program Studi Pendidikan Teknologi Informasi
Universitas Methodist Indonesia

lumbanrajaposma@gmail.com

ABSTRACT

One way to protect the copyright and authenticity of a digital image is to affix a watermark in the form of writing or a sign on the image. If the affixed watermark is visible, then its presence can interfere with the appearance of the image and it is possible that the watermark will be cut, discarded or cropped by parties who are not entitled to do so. In this paper, an application is designed to affix a watermark in the form of an invisible message, or an invisible watermark by using a combination of pixel value indicator and the Most Significant Bit (MSB) method on the image, so that it is not detected visually and the visual appearance of the image is not disturbed. The message is scrambled using a key, then the scrambled bits are hidden in the 5th bit MSB and the 6th bit of the green channel or blue channel, depending on the number of bits 1 owned by the red channel which serves as an indicator. The process of hiding the watermark is carried out in the middle area of the image, so it is difficult to cut to remove it. Applications designed can be used to affix an invisible watermark in the form of messages that are not easily removed on digital images, so that they can protect the copyright of a digital image.

Keywords: *Watermarking, Pixel Value Indicator, MSB*

I. PENDAHULUAN

Penambahan suatu label pada citra digital sering dilakukan oleh pemiliknya untuk berbagai tujuan. Penambahan label tersebut dikenal dengan *watermarking*. Yang menjadi permasalahan adalah *watermark* yang ditambahkan pada citra secara konvensional adalah *watermark* yang terlihat (*visible watermark*), sehingga *watermark* tersebut dapat dihilangkan oleh pihak lain. Sebagai contoh, penambahan *watermark* pada bagian atas atau bagian bawah citra menghadapi kemungkinan *watermark* dapat dipotong dan dihapus oleh pihak lain tanpa persetujuan pemiliknya. Kemudian apabila *watermark* semi transparan ditambahkan pada bagian tengah citra tentu akan mengganggu pengamatan visual terhadap citra. Untuk mengatasi masalah ini, dapat ditambahkan *invisible watermark* atau *watermark* yang tak terlihat secara kasat mata pada citra digital, sehingga dapat disisipkan informasi pada citra digital sesuai dengan yang diinginkan oleh pemilik citra. Salah satu metode *watermarking* yang sering digunakan adalah metode *Least Significant Bit (LSB)*, yaitu dengan menambahkan *bit watermark* pada bit terakhir piksel, sehingga perubahan warna yang diakibatkan sangat kecil dan tidak dapat dideteksi secara kasat mata. Namun metode ini sudah sangat umum dan para *hacker* sering dengan mudah menyerang dan mengekstraksi data, sehingga perlu dicari solusi untuk mengatasinya. Sebagai suatu langkah solusi, pada penelitian ini digunakan kombinasi *Pixel Value Indicator* dan metode *Most Significant Bit (MSB)* untuk menyisipkan pesan *watermark* pada citra. *Channel* warna

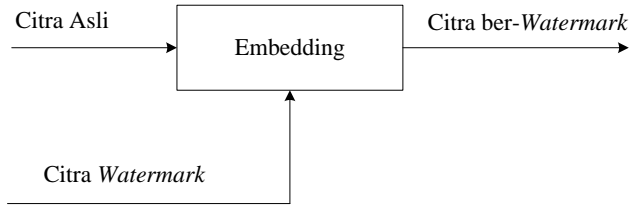
merah akan digunakan sebagai indikator untuk menyembunyikan data rahasia pada *MSB* bit ke-5 dan bit ke-6 dari *channel* warna hijau atau *channel* warna biru. Agar lebih aman, maka penyisipan *watermark* akan disertai dengan penggunaan kunci yang berfungsi sebagai *password* untuk mengekstrak *watermark* pada citra. Apabila kunci yang digunakan untuk ekstraksi, tidak sesuai, maka *watermark* tidak dapat diekstraksi. Selain itu, penggunaan kunci, penggunaan *channel* warna merah sebagai indikator, dan *MSB* bit ke-5 dan 6 *channel* warna hijau dan biru sebagai tempat penyimpanan data (*watermark*) merupakan skema baru dan akan meningkatkan keamanan *watermark*. Berdasarkan uraian di atas, maka dalam tulisan ini dibahas bagaimana proses untuk menyembunyikan *watermark* pada citra digital menggunakan kombinasi *pixel value indicator* dan metode *MSB*, yang diimplementasikan pada citra *.jpg, *.bmp dan *.png. hingga 999 karakter, password hingga 20 karakter, dapat menampilkan proses perhitungan yang singkat. Hasil akhir dari penelitian ini dapat digunakan untuk berbagai kepentingan, misalnya untuk menjamin keaslian citra, melindungi hak cipta, dan menghindari penyalahgunaan maupun pelanggaran hukum.

II. KERANGKA TEORI

Watermarking

Watermarking dapat dipandang sebagai suatu proses penyisipan sebuah data rahasia (*watermark*) terhadap sebuah gambar atau dokumen, yang berkembang hingga media digital atau disebut dengan *Digital Watermarking*

yang dapat dijalankan pada berbagai media digital, seperti citra digital, suara dan video. Secara umum, *watermarking* adalah proses penyisipan citra digital dengan pesan atau *watermark*, seperti terlihat pada gambar 1.



Gambar 1. Skema *Watermarking*

watermark yang disisipkan dalam citra digital haruslah *imperceptible* atau tidak terdeteksi oleh sistem penglihatan manusia (*Human Visual System*) atau sistem pendengaran manusia (*Human Auditory System*).

Metode *Least Significant Bit (LSB)*

Dalam metode ini, informasi disembunyikan dengan cara menyisipkannya pada bit rendah atau bit yang paling kanan (*LSB*) data pixel yang menyusun *file* tersebut. Untuk berkas bitmap 24 bit, setiap *pixel* (titik) pada gambar yang terdiri dari susunan tiga warna merah, hijau dan biru (*RGB*) masing-masing disusun oleh bilangan 8 bit (1 *byte*) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian, pada setiap *pixel* berkas bitmap 24 bit dapat disisipkan 3 bit data. Algoritma pada metode *LSB* lebih cepat dan lebih mudah. Misalkan, sebuah karakter "f" (kode *ascii* = 131) akan disisipkan ke dalam suatu citra *grayscale* 8 bit yang berukuran 10 x 10 *pixel*, seperti terlihat pada gambar 2 berikut.

Citra *grayscale* 8 bit, 10 x 10 *pixel*

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Gambar 2. Citra *Grayscale* 8 bit dengan ukuran 10 x 10 *pixel*

Langkah pertama adalah mengubah kedua data tersebut (huruf dan citra) menjadi nilai biner (gambar 2.3). Nilai biner untuk huruf "f" adalah 10000011. Oleh karena

jumlah digit biner huruf hanya 8 bit, maka jumlah *pixel* citra yang dibutuhkan untuk menyembunyikan huruf tersebut cukup 8 *pixel* saja.

8 <i>pixel</i> pertama diambil	Pixel Citra	Huruf "f" (<i>asc</i> =131)
1	1=00000001	1
6	6=00000110	0
5	5=00000101	0
3	3=00000011	0
7	7=00000111	0
4	4=00000100	0
7	7=00000111	1
4	4=00000100	1
0	0=00000000	
4	4=00000100	
4	4=00000100	
4	4=00000100	

Gambar 3. Pengubahan Nilai *Pixel* dan Huruf ke Biner

Langkah kedua adalah mengganti bit terakhir (*Least Significant Bit / LSB*) dari *pixel* citra dengan bit-bit dari huruf, seperti terlihat pada gambar 4 berikut.

Pixel citra	Huruf "f" (<i>asc</i> =131)	Pixel citra yang berubah
1=00000001	← 1	1=00000001
6=00000110	← 0	6=00000110
5=00000101	← 0	4=00000100
3=00000011	← 0	2=00000010
7=00000111	← 0	6=00000110
4=00000100	← 0	4=00000100
7=00000111	← 1	7=00000111
4=00000100	← 1	5=00000101

menjadi

Gambar 4. Proses Penyisipan dengan Metode *LSB*

Setelah penyisipan, bit-bit *pixel* citra yang mengalami perubahan hanya 4 *pixel* saja, seperti terlihat pada gambar 5 berikut.

4 *pixel* yang berubah

1	6	4	2	6	4	7	5	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Gambar 5. Hasil Penyisipan dengan Metode *LSB*

Pada gambar 5, terlihat bahwa piksel yang mengalami perubahan hanya sebesar 1 intensitas saja. Oleh karena itu, secara kasat mata, hal ini tidak terlalu berpengaruh, dan tidak semua piksel mengalami perubahan.

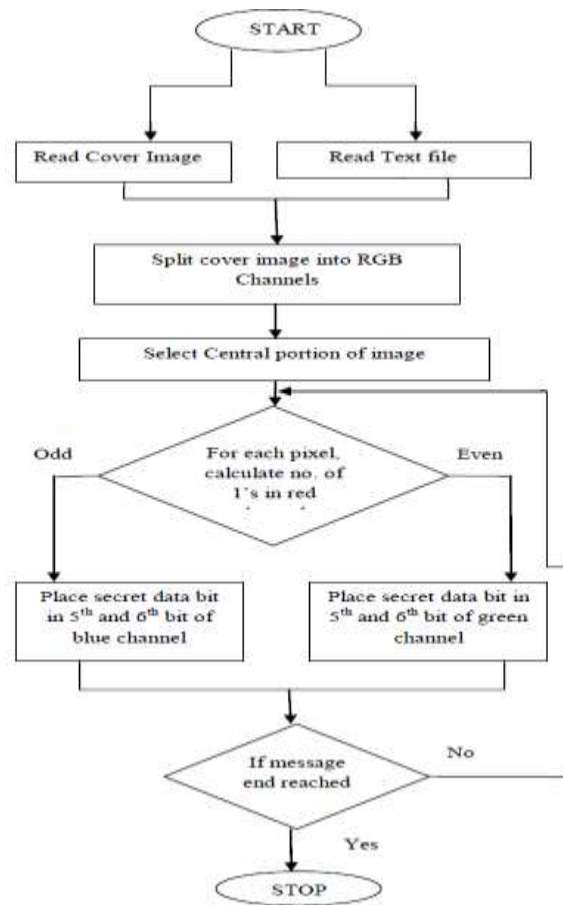
Kombinasi Pixel Value Indicator dan Most Significant Bit (MSB)

Kebalikan dari *LSB* adalah *Most Significant Bit (MSB)*. Pada metode *MSB*, bit-bit pesan disembunyikan pada bit-bit paling depan dari piksel citra. Namun, kelemahan dari metode *MSB* adalah warna citra berubah drastis, seperti terlihat pada gambar 6 berikut.



Gambar 6. Watermarking *LSB* dan *MSB*

Pada gambar 2.6 terlihat bahwa metode *LSB* tidak terlihat mengubah warna citra, bila dibandingkan dengan metode *MSB*. Namun perlu dilakukan modifikasi dengan menggunakan teknik *pixel indicator* untuk menyembunyikan *watermark* pada bit *MSB* citra. *Channel* warna merah akan digunakan sebagai indikator untuk menyembunyikan data rahasia pada *MSB* bit ke-5 dan bit ke-6 dari *channel* warna hijau atau *channel* warna biru disertai dengan penggunaan kunci yang berfungsi sebagai *password* untuk mengekstrak *watermark* dari citra, sehingga *watermark* lebih sulit diekstraksi. Penggunaan kunci, penggunaan *channel* warna merah sebagai indikator, dan *MSB* bit ke-5 dan 6 *channel* warna hijau dan biru sebagai tempat penyimpanan data (*watermark*) merupakan suatu tindakan yang akan meningkatkan keamanan *watermark*. *Flowchart* dari penyisipan bit pesan pada citra digital dapat dilihat pada gambar 7.



Gambar 7. Proses Penyembunyian Pesan pada Citra

Berikut adalah algoritma yang digunakan untuk mengekstraksi pesan dari citra:

1. Pilih gambar yang telah memiliki *watermark*.
2. Temukan bagian tengah citra.
3. Bagi citra menjadi 3 *channel*, yaitu *Red*, *Green*, dan *Blue*.
4. Untuk setiap piksel 24 bit, lakukan langkah no. 5 dan 6.
5. Jika jumlah bit 1 pada *channel Red* adalah genap, maka ekstraksi bit pesan dari posisi bit ke-5 dan bit ke-6 dari *channel Green*.
6. Jika jumlah bit 1 pada *channel Red* adalah ganjil, maka ekstraksi bit pesan dari posisi bit ke-5 dan bit ke-6 dari *channel Blue*.
7. *Output* adalah pesan *watermark*.

III. PEMBAHASAN

Proses pembahasan mencakup proses penentuan daerah penyembunyian *watermark*, proses penyembunyian *watermark* dengan kombinasi *Pixel Value Indicator* dan metode *Most Significant Bit (MSB)*, proses ekstraksi *watermark*, dan pemodelan sistem.

a. Penentuan Daerah Penyembunyian *Watermark*

Pada penelitian ini, *watermark* disembunyikan di area tengah citra, dengan pertimbangan bahwa area tengah merupakan inti atau sentral dari citra dan jauh lebih aman dari proses *cropping*

Dengan menggunakan kombinasi *pixel value indicator* dan metode *MSB*, satu piksel citra dapat menampung 2 bit biner *watermark* pada *channel* warna biru atau *channel* warna hijau, bergantung pada indikator *channel* warna merah. Batas *watermark* yang dapat disembunyikan pada citra dibatasi sebanyak 999 karakter, bila diubah ke bit biner adalah sebesar 999 x 8 atau sama dengan 7992 bit biner. Apabila 7992 bit dibagi dengan 2 bit, maka dibutuhkan 3996 piksel untuk menampung seluruh karakter *watermark*. Untuk menentukan luas area piksel yang menampung 3996 piksel, maka dilakukan

perhitungan $\sqrt{3996}$ atau didapat angka 63.21 atau dibulatkan menjadi 64. Dengan demikian, dibutuhkan area seluas 64 x 64 piksel di area tengah citra untuk menampung karakter *watermark*. Sebagai contoh, digunakan citra berukuran 640 x 480 piksel, seperti terlihat pada gambar 8.



Gambar 8. Citra Ukuran 640 x 480 Piksel

Dengan ukuran 640 x 480 piksel, maka perhitungan posisi awal penyembunyian *watermark* dimulai pada posisi:

Ukuran citra : Lebar = 640 Tinggi = 480

Piksel di tengah citra : X = 640 / 2 = 320
Y = 480 / 2 = 240

Ukuran area *watermark* : 64 x 64 piksel

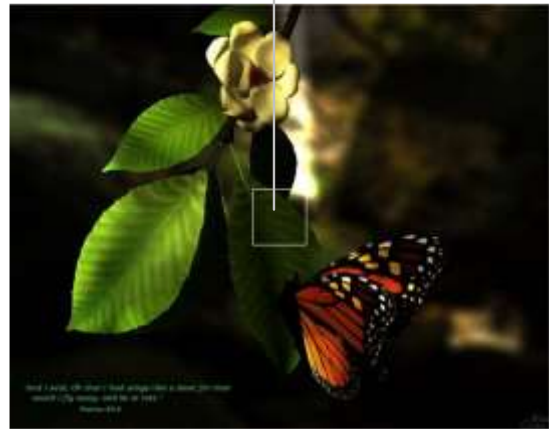
Posisi tengah area citra : X = (Lebar citra / 2) – (Uk. area *watermark*/2)
Y = (Tinggi citra / 2) – (Uk

area *watermark*/2)

Posisi tengah area citra : X = 320 – (64/2) = 288
Y = 240 – (64/2) = 208

Dengan demikian untuk citra berukuran 640 x 480 piksel, area penyembunyian data *watermark* akan dimulai pada posisi piksel X = 288 dan Y = 208, seperti terlihat pada gambar 9.

Area watermark 64 x 64 piksel



Gambar 9. Area Penyembunyian *Watermark*

b. Penyembunyian *Watermark*

Berikut adalah algoritma yang digunakan untuk menyembunyikan *watermark* pada citra dengan kombinasi *pixel value indicator* dan metode *MSB*:

1. *Input* citra untuk diberikan *watermark*.
2. *Input* kunci dan *watermark*.
3. Pilih bagian tengah citra
4. Piksel ke-1 dan piksel ke-2 pada area tengah citra dimasukkan penanda dimulainya *watermark*, yaitu: “RIDOCG” (setelah di-*XOR* dengan kunci), sedangkan piksel ke-3 dimasukkan jumlah karakter *watermark*.
5. *Watermark* diproses dengan fungsi *XOR* dengan kunci rahasia.
6. Untuk setiap piksel 24 bit, lakukan langkah berikut:
 - a. Jika jumlah bit 1 pada *channel Red* adalah genap, maka tempatkan dua bit *watermark* pada posisi bit ke-5 dan bit ke-6 dari *channel Green*.
 - b. Jika jumlah bit 1 pada *channel Red* adalah ganjil, maka tempatkan dua bit *watermark* pada posisi bit ke-5 dan bit ke-6 dari *channel Blue*.
7. *Output* adalah citra yang telah disisipi pesan *watermark*.

c. Ekstraksi *Watermark*

Berikut adalah algoritma yang digunakan untuk mengekstraksi *watermark* dari citra dengan kombinasi *pixel value indicator* dan metode *MSB*:

1. *Input* citra hasil *watermark*.
2. *Input* kunci.
3. Cari penanda awal pada citra sebagai posisi awal area *watermark*.
4. Ambil besarnya karakter *watermark* yang akan diekstraksi pada penanda.
5. Untuk setiap piksel 24 bit, lakukan langkah berikut:
 - a. Jika jumlah bit 1 pada *channel Red* adalah genap, maka ekstraksi dua bit *watermark* pada posisi bit ke-5 dan bit ke-6 dari *channel Green*.

- b. Jika jumlah bit 1 pada *channel Red* adalah ganjil, maka ekstraksi dua bit *watermark* pada posisi bit ke-5 dan bit ke-6 dari *channel Blue*.
- 6. Bit ekstraksi diproses dengan fungsi *XOR* dengan kunci rahasia.
- 7. *Output* hasil *XOR* adalah karakter *watermark*.

Perancangan Sistem

Aplikasi implementasi *watermarking* pada citra digital menggunakan kombinasi *Pixel Value Indicator* dan metode *Most Significant Bit* untuk proteksi hak cipta citra digital ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Basic .Net 2008*. Aplikasi mempunyai beberapa *form*, diantaranya adalah: *Form* Utama, *Form* Penyembunyian *Watermark*, *Form* Ekstraksi *Watermark*, *Form* Perbandingan Citra, dan *Form* About.

1. Form Utama

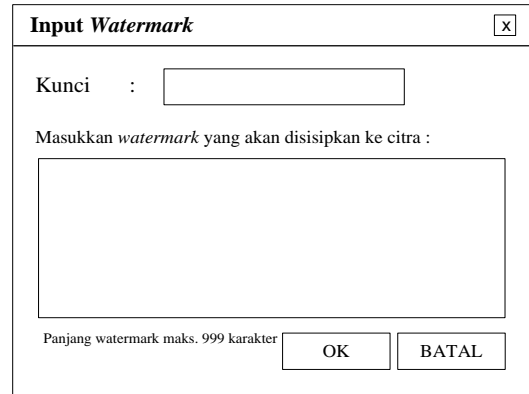
Pada *form* utama, terdapat *toolbar* untuk melakukan proses penyembunyian dan ekstraksi *watermark*. Rancangan *form* utama dapat dilihat pada gambar 10.



Gambar 10. Rancangan *Form* Utama

2. Form Penyembunyian Watermark

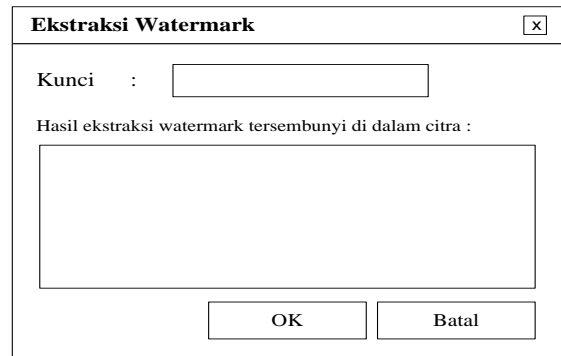
Pada *form* ini, aplikasi akan meminta *input* kunci dan *input watermark*, sesaat setelah pengguna mengklik tombol ‘Penyembunyian *Watermark*’ pada *form* Utama. Tujuannya adalah agar aplikasi dapat menyembunyikan *watermark* ke citra dengan menggunakan kunci. Rancangan *form* dapat dilihat pada gambar 11.



Gambar 11. Rancangan *Form* Penyembunyian *Watermark*

3. Form Ekstraksi Watermark

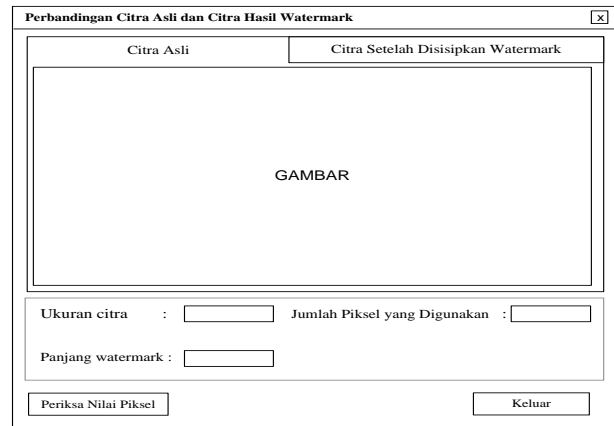
Form ini muncul setelah pengguna mengklik tombol ‘Ekstraksi *Watermark*’ dari *form* utama. Fungsi *form* ini adalah untuk memasukkan kunci dan menampilkan hasil ekstraksi *watermark*. Rancangan *form* dapat dilihat pada gambar 12.



Gambar 12. Rancangan *Form* Ekstraksi *Watermark*

4. Form Perbandingan

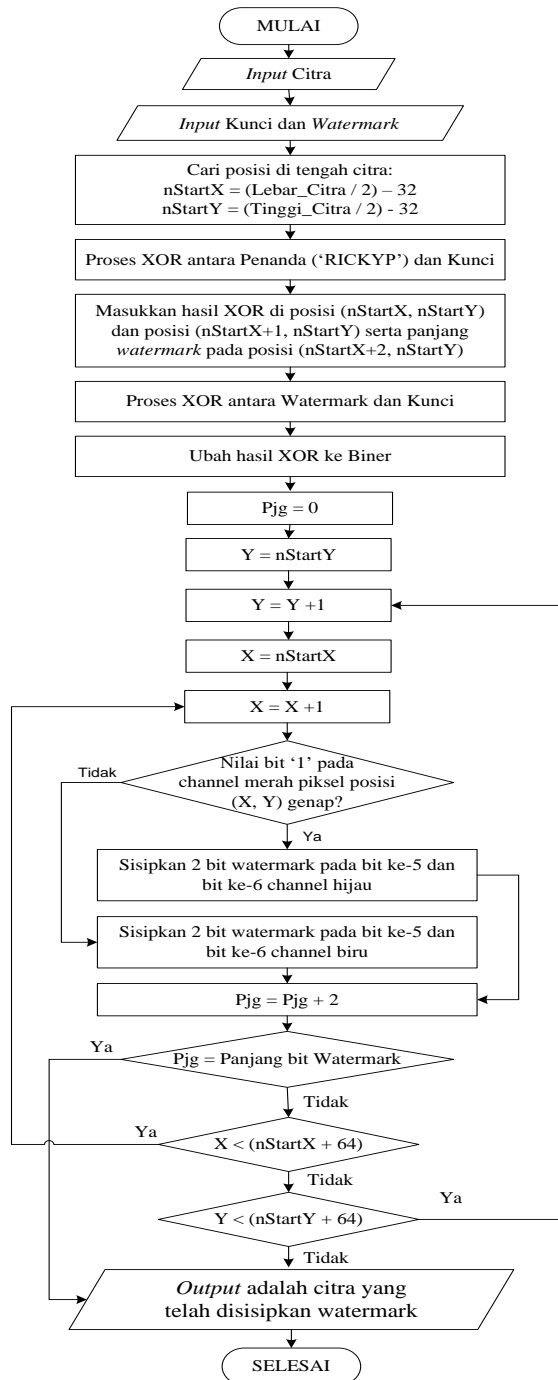
Pada *form* ini, pengguna dapat membandingkan tampilan citra asli dan citra hasil *watermark*. Rancangan *form* dapat dilihat pada gambar 13.



Gambar 13. Rancangan *Form* Perbandingan Citra

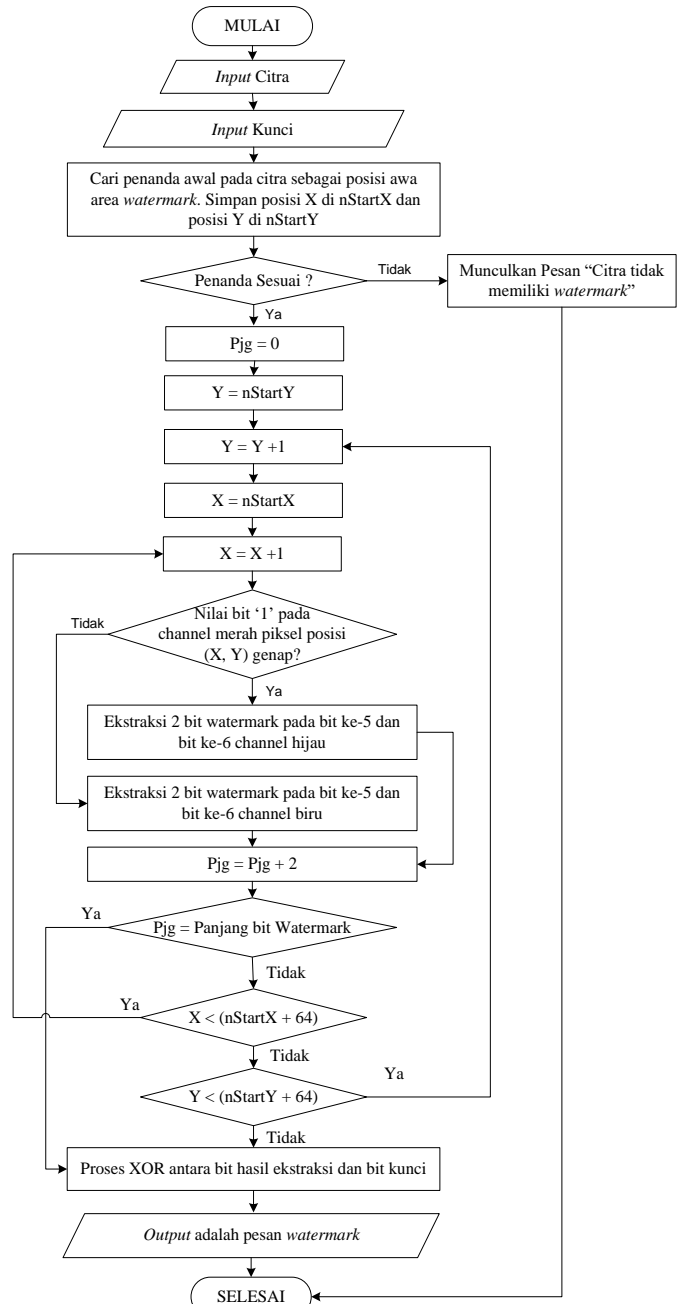
IV. ALGORITMA DAN IMPLEMENTASI

Algoritma dari proses penyembunyian *watermark* pada citra dengan kombinasi *pixel value indicator* dan metode *MSB* dapat digambarkan dalam bentuk *flowchart*, seperti terlihat pada gambar 15.



Gambar 15. *Flowchart* Algoritma Penyembunyian *Watermark* pada Citra

Proses ekstraksi *watermark* dengan kombinasi *pixel value indicator* dan metode *MSB* dapat dilihat pada *flowchart*, seperti terlihat pada gambar 16.



Gambar 16. *Flowchart* Algoritma Ekstraksi *Watermark* dari Citra

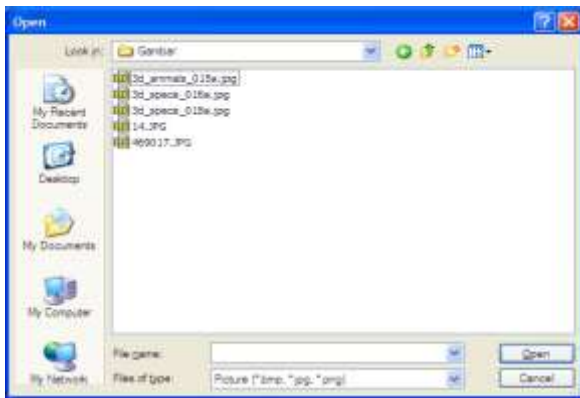
Implementasi

Berikut dijelaskan mengenai implementasi dari aplikasi *watermarking* pada citra digital menggunakan kombinasi *Pixel Value Indicator* dan metode *Most Significant Bit* untuk proteksi hak cipta citra digital. *Form* Utama akan muncul pada saat aplikasi dijalankan, seperti terlihat pada gambar 17.



Gambar 17. Form Utama

Untuk membuka gambar, klik tombol “Buka Gambar” pada *toolbar* di bagian atas *form* utama. Kotak dialog *Open File* dari *windows* akan muncul seperti terlihat pada gambar 18.



Gambar 18. Kotak Dialog *Open File*

Pilih salah satu gambar dan klik tombol “Open”, dan *file* gambar akan di-*load* pada *form* utama, seperti terlihat pada gambar 19.



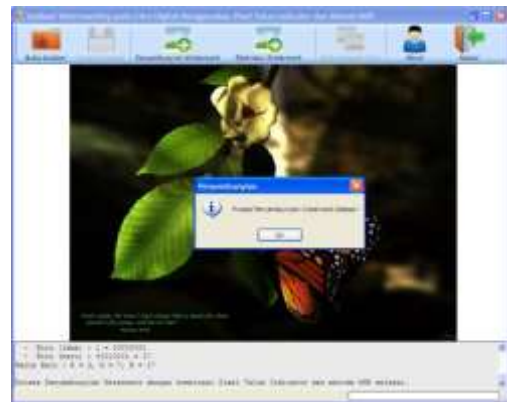
Gambar 19. Tampilan Gambar pada *Form* Utama

Untuk menyembunyikan *watermark* pada citra, tekan tombol “Penyembunyian *Watermark*” pada *toolbar* di bagian atas *form* utama, dan *form* penyembunyian *watermark* akan muncul seperti terlihat pada gambar 20.



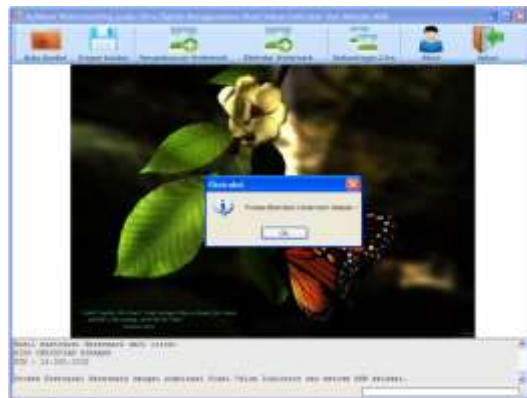
Gambar 20. Form Penyembunyian *Watermark*

Masukkan kunci atau *password* (sebagai contoh “kunci12345”), dan *watermark* yang akan disisipkan, lalu tekan tombol “OK” untuk memulai proses penyembunyian *watermark* ke citra digital, seperti terlihat pada gambar 21.



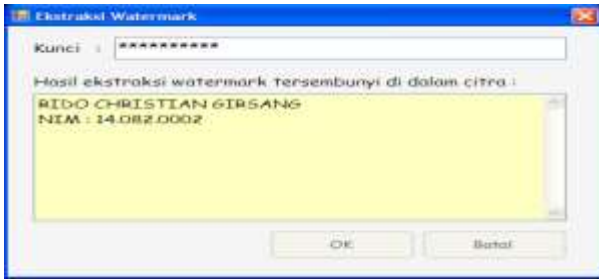
Gambar 21. Proses Penyembunyian *Watermark*

Untuk melakukan ekstraksi *watermark* dari citra, tekan tombol “Ekstraksi *Watermark*”, dan proses ekstraksi akan berjalan, seperti terlihat pada gambar 22.



Gambar 22. Proses Ekstraksi *Watermark*

Dengan menggunakan kunci yang benar, maka *watermark* akan terekstraksi dengan benar pada *form* Ekstraksi *Watermark*, seperti terlihat di gambar 23.



Gambar 23. Hasil Ekstraksi *Watermark* dengan Kunci Benar

Apabila proses ekstraksi *watermark* menggunakan kunci yang salah, maka *watermark* tidak dapat diekstraksi, seperti terlihat pada gambar 24.



Gambar 24. Proses Ekstraksi *Watermark* Kunci yang Salah

Untuk melihat perbandingan antara citra asli dengan citra hasil *watermark*, maka tekan tombol “Perbandingan Citra”, dan *form* perbandingan akan muncul seperti terlihat pada gambar 4.11. *Form* perbandingan berisi citra asli, citra setelah disisipkan *watermark*, ukuran citra, panjang *watermark*, dan jumlah piksel yang digunakan pada citra untuk menyembunyikan *watermark*. Apabila dibandingkan secara manual, citra hasil penyembunyian *watermark* terlihat sama dengan citra asli. Akan tetapi terdapat perbedaan nilai warna piksel yang tidak terlihat secara kasat mata. Untuk dapat melihat perbandingan nilai piksel antara citra asli dan citra hasil *watermark*, maka *user* dapat menekan tombol “Periksa *Watermark*” untuk memeriksa nilai warna piksel pada posisi piksel yang ditentukan.



Gambar 25. *Form* Perbandingan Citra

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah menyelesaikan perancangan Aplikasi implementasi *watermarking* pada citra digital menggunakan kombinasi *Pixel Value Indicator* dan metode *Most Significant Bit*, beberapa hal yang dapat disimpulkan dari penelitian adalah sebagai berikut:

1. Aplikasi dapat digunakan untuk melindungi hak cipta sebuah citra dengan menyembunyikan *watermark* pada citra digital menggunakan kombinasi *Pixel Value Indicator* dan metode *MSB*, yaitu dengan menyembunyikan data rahasia pada *MSB* bit ke-5 dan bit ke-6 dari *channel* warna hijau atau *channel* warna biru menggunakan *channel* warna merah sebagai indikator.
2. Aplikasi menggunakan kunci rahasia, sehingga apabila proses ekstraksi menggunakan kunci yang salah, maka *watermark* tidak dapat diekstraksi.
3. Citra hasil *watermark* terlihat sama dengan citra asli secara kasat mata, walaupun sebenarnya terdapat perbedaan nilai piksel.
4. *Watermark* disembunyikan pada bagian tengah dari citra, sehingga *watermark* tetap dapat diekstraksi walaupun citra dipotong (*crop*) pada bagian atas, bagian bawah, samping kiri atau samping kanan.

Saran

Untuk pengembangan aplikasi lebih lanjut di masa yang akan datang, beberapa hal yang dapat disarankan sebagai berikut:

1. Disarankan untuk menambahkan algoritma kompresi pada pesan *watermark*, sehingga dapat memperkecil ukuran pesan *watermark* yang akan disembunyikan.
2. Disarankan untuk mengembangkan aplikasi menjadi aplikasi steganografi yang dapat menyembunyikan *file* rahasia pada citra digital.

6. DAFTAR PUSTAKA

- [1]. Anand, K, and Sharma, M, 2014, *Data Security Using LSB & MSB Image Steganography*, Haryana, India : Swami Devi Dyal Institute of Engg & Technology
- [2]. Hakim, A.R, 2012, *Analisa Perbandingan Watermarking Image Menggunakan Discrete Wavelet Transform*, Universitas Indonesia, Depok
- [3]. Hidayatullah, P , 2017, *Pengolahan Citra Digital (Teory dan AplikasinNyata)* Informatika, Bandung
- [4]. Paradhipta, B.R,2015, *Penerapan Teknik Watermarking Metode DWT pada Citra Digital*, Universitas Sanata Dharma, Yogyakarta.
- [5]. Sharma, A., 2017, *A Secure SteganographyTechnique Using MSB*, Kurukshetra University, India