Fragile Image Watermarking yang Aman untuk Verifikasi Gambar Menggunakan Chaotic Arnold's Cat Map dan Blum Blum Shub

Secure Fragile Image Watermarking for Image Verification using Chaotic Arnold's Cat Map and Blum Blum Shub

Marcellus Michael Herman Kahari - 13520057 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): michaelkahari.mk@gmail.com

***Abstract*—Di tengah zaman yang semakin berkembang, semakin sulit menentukan apakah suatu file ditulis oleh yang berhak atau tidak. Dengan menggunakan *fragile watermarking*, pengguna dapat memastikan apakah suatu gambar diubah oleh pihak yang tidak bertanggung jawab atau tidak. Pada makalah ini, akan dibahas *fragile watermarking* menggunakan fungsi *chaos* Arnold's Cat Map. Tujuan dari penggunaan fungsi *chaos* ini adalah mengacak *digital watermarking* sehingga tidak dapat dikenali oleh peretas. Jumlah iterasi pada Arnold's Cat Map didapatkan menggunakan pembangkit bilangan acak dengan masukan bits pada gambar yang digunakan untuk verifikasi *fragile watermarking*.**

Keywords— Fragile Watermarking, Chaotic, Arnold's Cat Map, Blum Blum Shub

I. Pendahuluan

Integritas data merupakan isu yang penting untuk ditelaah zaman ini. Cukup banyak kakas-kakas yang dapat digunakan untuk melakukan modifikasi suatu data sedemikian rupa sehingga penerima tidak menyadari jika data telah diubah. Hal ini tentu mengancam keamanan privasi data seseorang, terlebih jika data tersebut disalahgunakan seperti untuk menjatuhkan nama baik seseorang.

Salah satu data yang sering dan mudah dimodifikasi adalah gambar. Banyak kakas daring yang dapat digunakan untuk melakukan modifikasi pada suatu gambar, seperti memotong, menempel suatu gambar lain, mengubah kecerahan, dan lain sebagainya. Jika gambar tersebut bukanlah suatu gambar yang penting, hal ini tentu bukan masalah penting. Akan tetapi, jika gambar tersebut mengandung informasi-informasi penting, seperti gambar kartu tanda penduduk (KTP), tentu besar kemungkinan terjadi penyalahgunaan modifikasi gambar.

Untuk mengatasi permasalahan ini, digunakan *fragile watermarking* guna memastikan dan memverfikasi suatu gambar. *Fragile watermarking* ditujukan untuk menjaga integritas dan orisinalitas suatu gambar. Cara kerja dari *fragile watermarking* adalah dengan menyisipkan *watermark* pada suatu gambar menggunakan metode *least significant byte* (LSB). *Fragile watermarking* sensitif terhadap segala perubahan citra pada gambar, seperti perubahan kontras, penambahan gambar, hingga penambahan teks.

Selain *fragile watermarking*, terdapat *robust watermarking*. *Robust watermarking* digunakan untuk menyisipkan label kepemilikan pada suatu citra digital. Pada *robust watermarking*, label kepemilikan sangat sulit untuk dihilangkan walaupun telah dilakukan perubahan kontras, pemotongan citra, dan lain sebagainya. Akan tetapi, karena pada makalah ini hanya dibahas mengenai keaslian dan integritas data, digunakan *fragile watermarking*.

Namun, skema *fragile watermarking* biasa kurang aman. Hal ini disebabkan terdapat peluang orang yang tidak berhak melakukan autentikasi. Untuk menghindari peluang orang yang tidak berhak melakukan autentikasi terhadap gambar, digunakan skema fungsi *chaos* pada Arnold's Cat Map. Hal ini akan menyebabkan sedikit saja perubahan data untuk melakukan verifikasi, sistem akan mendeteksi kesalahan.

Selain itu, guna semakin memperkuat Arnold's Cat Map, digunakan algoritma pembangkit bilangan acak atau *cryptographically secure pseudorandom generator* (CSPRNG) Blum Blum Shub (BBS). Algoritma pembangkit bilangan acak digunakan untuk membangkitkan jumlah iterasi Arnold's Cat Map dengan masukan bit pada gambar *watermark*. Jika *watermark* berubah sedikit saja, algoritma pembangkit bilangan acak ini akan mendeteksi hasil masukan yang berbeda dan menyebabkan iterasi Arnold's Cat Map berubah jumlahnya.

II. Dasar Teori

A. Fragile Image Watermarking

Menurut Ganic, E, Zunair, N, dan Eskicioglu, A.M., (2003) watermark merupakan teknik penyisipan data ke dalam elemen multimedia seperti citra, audio atau video. Watermark memiliki tujuan untuk melakukan perlindungan *copyright*, pembuktian kepemilikan, serta autentikasi.

Melakukan penyisipan watermark ke dalam citra digital dapat dilakukan melalui dua cara, yaitu *visible watermarking*

dan *invisible watermarking*. Terdapat dua klasifikasi *invisible image watermarking*, yaitu *fragile watermarking* dan *robust watermarking*. Pada makalah ini, akan digunakan *fragile watermarking* dalam implementasi verifikasi gambar.

Fragile image watermarking digunakan untuk menjaga ketahanan gambar dari perubahan pada citra digital. Watermark akan menjadi rusak atau pecah jika dilakukan manipulasi. Metode yang digunakan untuk melakukan implementasi fragile image watermarking adalah least significant byte (LSB).

LSB bekerja dengan cara menyisipkan pesan pada gambar yang telah diubah menjadi format byte, kemudian menambahkan bit terakhir dari byte tersebut dengan pesan yang hendak disembunyikan. Penambahan pada umumnya cenderung kecil sehingga cukup sulit dibuktikan oleh mata manusia.

Alur kerja dari fragile image watermarking adalah sebagai berikut.

1. Membuat watermark seukuran dengan citra yang akan disisipi, bisa dilakukan dengan melakukan *copy paste*.
2. Watermark akan disisipkan menggunakan metode LSB
3. Ekstrasi watermark dilakukan dengan mengambil bit-bit LSB pada setiap pixel citra, lalu disatukan sehingga menjadi gambar watermark semula.

B. Arnold's Cat Map

Arnold's Cat Map (ACM) adalah fungsi *chaos* yang bersifat *reversible*. Dilansir dari wolfram mathworld, ACM adalah fungsi non-Hamiltonian, nonanalytic, dan mixing. ACM melakukan transformasi koordinat pada suatu citra dari (x,y) menjadi (x',y'). Fungsi transformasi yang digunakan adalah sebagai berikut.

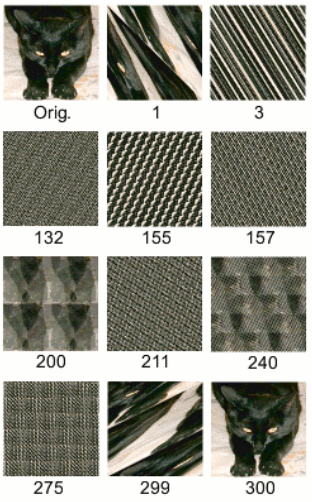
$$\Gamma(x, y) = (x + by,\ cx + bcy + y)\ mod\ 1$$

B dan c pada fungsi merupakan integer sembarang dan jika fungsi dipetakan menjadi bentuk matriks, matriks tersebut harus memiliki determinan yang bernilai 1. Tujuan dari syarat ini adalah agar hasil transformasinya bersifat *area-preservering*.

Nilai b dan c dapat dikatakan sebagai nilai yang bersifat rahasia. Selain itu, nilai m pada ACM juga bersifat rahasia. M adalah variabel yang menentukan jumlah putaran ACM yang diimplementasikan kepada citra.

Dikutip dari We-bin (2009), setelah ACM dilakukan iterasi sebanyak m kali, terdapat T sedemikian sehingga dipenuhi persamaan berikut $(x_T, y_T) = (x, y)$ untuk nilai T yang bergantung terhadap b, c, dan N dengan N adalah ukuran citra.

ACM adalah fungsi yang *reversible*. Misalkan T adalah periode dari suatu citra ACM dan menyatakan jumlah iterasi atau m sejak gambar mulai dilakukan transformasi hingga gambar kembali ke bentuk semula. Menurut Falk, Harold (1992), jumlah T tidak akan melebihi 3N dengan N adalah ukuran citra.



Ilustrasi hasil lelaran ACM

C. CSPRNG

Cryptographically secure pseudorandom generator (CSPRNG) atau dapat disebut sebagai pembangkit bilangan acak kriptografi yang khusus diciptakan untuk bidang kriptografi. CSPRNG bersifat deterministik, yaitu bilangan acak dapat dibangkitkan kembali jika menggunakan umpan yang sama.

Terdapat dua syarat suatu pembangkit bilangan acak dapat disebut sebagai CSPRNG.

1. Lolos uji keacakan secara statistik (randomness test).
2. Tahan terhadap serangan yang serius.

D. Blum Blum Shub

Blum Blum Shub (BBS) adalah salah satu algoritma CSPRNG yang dapat dikatakan praktif dan efisien secara kompleksitas algoritma. Algoritma ini didasarkan pada penggunaan teori bilangan. Keluaran dari algoritma ini adalah barisan bilangan acak berupa bit-bit biner. Bentuk dari BBS adalah sebagai berikut.

$$X_{n+1} = X_n^{\ 2} mod\ m$$

m adalah suatu bilangan yang didapatkan dari perkalian antara p dan q dengan p dan q adalah suatu bilangan prima yang dirahasakan dan kongruen dengan 3 (mod 4). Dipilih umpan untuk melakukan perhitungan pertama kali, misalkan s, dengan syarat lebih besar atau sama dengan 2 dan kurang dari m. Untuk setiap perhitungan, diambil LSB dan LSB tersebut digunakan sebagai bilangan acak yang dibangkitkan.

BBS dapat dikatakan sebagai CSPRNG yang aman, sebab sangat sulit untuk membedakan bit-bit luaran yang dihasilkan secara acak, dengan kesulitan paling sedikit setara dengan kesulitan untuk memecahkan persoalan *quadratic residue problem*. Selain itu, sulit untuk melakukan pemfaktoran nilai m yang cukup besar. Nilai m ini dapat disebarkan kepada khalayak umum.

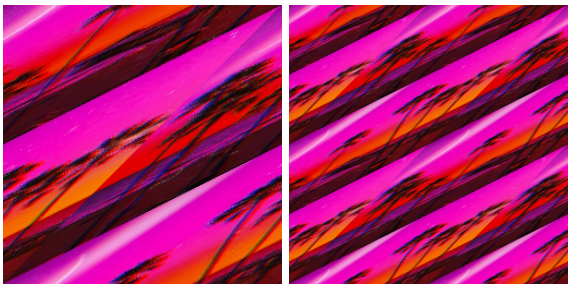
III. Implementasi

Program akan menerima dua buah masukan, yaitu gambar utama yang akan ditambahkan watermark dan watermark. Ukuran dari kedua gambar bisa berbeda. Proses pertama adalah dilakukan pembuatan watermark random. Watermark random dibuat dari gambar watermark yang dirandom dengan Arnold's Cat Map sebanyak M kali, dengan M kali didapatkan secara acak dari CSPRNG Blum Blum Shub.

Gambar watermark dibaca terlebih dahulu nilai LSB nya, kemudian dari nilai LSB tersebut akan dijadikan input masukan pertama kali pada BBS. Cara kerjanya adalah dengan melakukan iterasi setiap pixel pada bit gambar. Jika nilai LSB nya ganjil, nilai akan ditambah 1, dan jika genap tidak ditambah apapun. Hal ini akan menjadikan nilai m keluaran dari BBS akan sangat bergantung terhadap gambar watermark. Jika gambar watermark berubah sedikit, hasil random dari watermark akan berubah jauh. Berikut adalah ilustrasi dari percobaan hasil random watermark normal.

Berikut adalah ilustrasi dari percobaan hasil random watermark yang telah diubah warna kontrasnya. Dapat dilihat bahwa hasil random watermark ini berbeda jauh dibandingkan dengan gambar sebelumnya. Hal ini menunjukkan bahwa perubahan sedikit saja di watermark akan menghasilkan gambar random yang sangat berbeda.

Setelah gambar watermark dilakukan perandoman, ukuran dari watermark akan disamakan dengan gambar utama. Cara melakukan penyamaan ini adalah dengan melakukan duplikasi gambar hingga ukuran gambar watermark lebih besar dari ukuran gambar utama. Setelah ukuran gambar watermark lebih besar, gambar watermark akan dipotong sedemikian sehingga berukuran sama dengan gambar utama. Berikut adalah ilustrasi dari duplikasi gambar hasil random.



Setelah ukuran dari watermark sama dengan ukuran dari gambar utama, gambar watermark akan disisipkan ke gambar utama dengan metode LSB. Penyisipan ini tidak begitu terlihat di mata manusia karena hanya memiliki selisih nilai bit yang kecil dengan gambar semula.

Gambar watermark yang disisipkan pada gambar utama dapat dibangkitkan kembali dengan mengambil nilai LSB dari gambar utama. Gambar watermark kemudian akan ditampilkan di layar serta program akan memberi tahu apakah watermark yang dibangkitkan dari gambar utama sama dengan masukan watermark yang dimasukkan pengguna. Jika sama, hal ini menunjukkan bahwa gambar utama tidak mengalami perubahan dan modifikasi apapun.

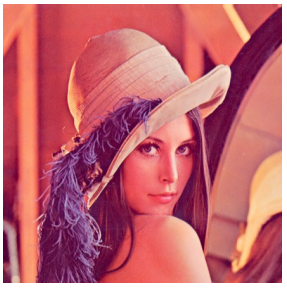
Program ini mengimplementasikan fragile image watermarking. Hal ini menyebabkan perubahan sekecil apapun dapat dideteksi oleh sistem. Perubahan-perubahan yang akan dilakukan pengujian pada bagian berikutnya adalah sebagai berikut:

1. Penambahan teks
2. Penambahan gambar
3. Pengubahan warna dan kontras
4. Pengubahan tipe file citra

IV. Uji Coba Program

Setelah program berhasil diimplementasikan dengan baik, akan dilakukan uji coba program dengan berbagai kasus uji. Tujuan dari pengujian ini adalah untuk membuktikan bahwa watermark yang terpasang pada program akan rusak jika dilakukan pengubahan-pengubahan pada gambar.

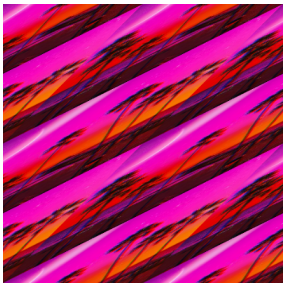
Uji coba akan menggunakan gambar Lenna sebagai gambar utama yang berukuran 440 x 440 pixel. Gambar watermark yang digunakan adalah gambar sebuah pemandangan yang berukuran 256 x 256 pixel.



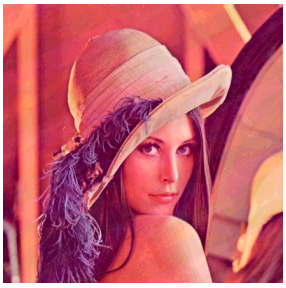
Gambar target

Gambar watermark

Untuk algoritma Blum Blum Shub, digunakan bilangan prima p = 7 dan q = 31 sehingga nilai n adalah 217. Nilai p dan q sudah memenuhi ke-kongruenan 3 mod 4. Jumlah iterasi algoritma BBS didapatkan dari nilai LSB pada watermark.



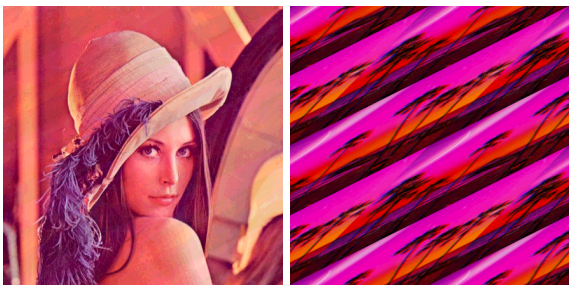
Gambar Watermark yang telah dirandom sebagai gambar pembanding



Gambar Lenna Watermark

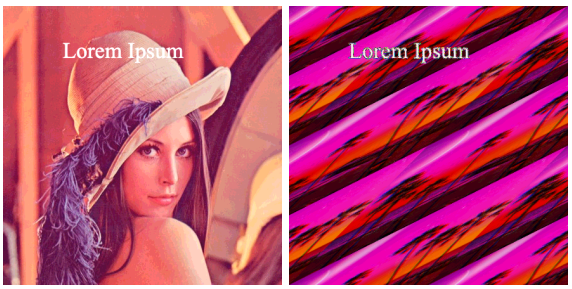
A. Kasus Normal

Kasus yang pertama adalah kasus normal. Berikut akan ditampilkan hasil citra dari Lenna yang telah disisipkan watermark random dan watermark yang dibangkitkan dari citra tersebut. Dapat dilihat dari hasil bahwa watermark yang dibangkitkan tidak memiliki perbedaan dengan gambar pembanding.



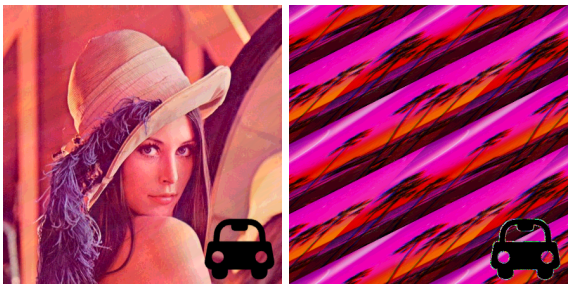
B. Kasus Penambahan Teks

Pada kasus uji ini, akan ditambahkan teks pada gambar Lenna yang telah disisipkan watermark. Teks yang ditambahkan adalah teks berwarna putih bertuliskan Lorem Ipsum. Teks ini akan ditambahkan pada tengah-tengah gambar. Penambahan teks ini tentu akan mengubah bit pada gambar sehingga ketika dibangkitkan, pada watermark justru akan muncul tulisan dari teks yang ditambahkan seperti pada gambar berikut.



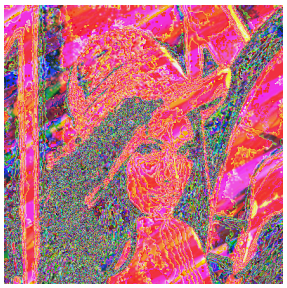
C. Kasus Penambahan Gambar

Kasus uji ketiga adalah penambahan gambar. Sama seperti penambahan teks, penambahan gambar akan mengubah nilai bit pada citra sehingga ketika dibangkitkan, akan terdapat bentuk gambar pada watermark yang dibangkitkan. Hal ini menunjukkan bahwa manipulasi gambar akan dapat dengan mudah terdeteksi pada gambar yang telah dilakukan watermarking.



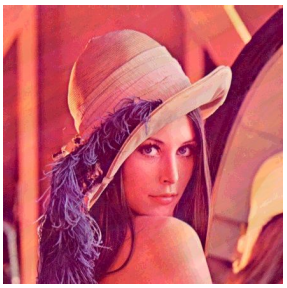
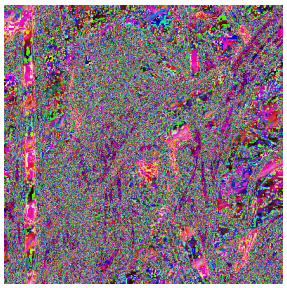
D. Kasus Pengubahan Warna dan Kontras

Kasus uji keempat adalah pengubahan warna dan kontras pada citra. Pada kasus uji ini, gambar Lenna yang telah disispkan watermark akan diubah kontras warnanya. Hal ini tentu akan berdampak pada bit-bit pada gambar. Karena perubahan ini terjadi secara menyeluruh, watermark yang dibangkitkan akan tampak rusak dan sangat jauh berbeda dengan gambar pembanding. Hal ini tentu cukup berbeda dengan dua kasus uji sebelumnya dimana perbedaan watermark yang dibangkitkan dengan gambar pembanding hanya terjadi pada sebagian watermark saja. Berikut adalah hasil pengubahan dan gambar watermark yang dibangkitkan.

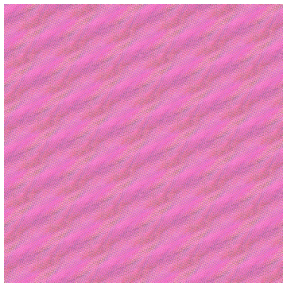
E. Kasus Pengubahan Tipe File

Kasus uji kelima adalah pada gambar Lenna yang telah disispkan watermark, dilakukan pengubahan tipe file. Tipe file yang digunakan pada gambar ber-watermark adalah png. Kemudian, dilakukan pengubahan tipe file dari png menjadi jpg. Setelah itu, watermark dibangkitkan dari gambar tersebut. Dapat dilihat pada hasil uji coba, watermark yang dihasilkan sangat berantakan dan berbeda dengan gambar pembanding. Hal ini menunjukkan bahwa fragile image watermarking juga tahan terhadap perubahan tipe file. Berikut adalah hasil pembangkitkan watermark.

F. Kasus Pengubahan Watermark Awal

Kasus uji keenam adalah melakukan pengubahan warna dan kontras pada watermark. Pengubahan ini bertujuan untuk membuktikan bahwa jika watermark dilakukan pengubahan sedikit saja, hasil gambar watermark acak yang dihasilkan jauh berbeda. Berikut adalah gambar dengan watermark yang warna dan kontras diubah, dan gambar hasil random watermark yang diubah.

G. Kasus Pengecekan dengan Watermak Random yang Diubah

Kasus uji ketujuh berbeda dengan kasus uji keenam. Pada kasus uji ketujuh gambar yang dilakukan modifikasi adalah gambar watermark yang telah dirandom untuk verifikasi. Gambar watermark ini diubah warna dan kontrasnya.

Setelah dilakukan pengubahan, gambar watermark kemudian dibangkitkan dari gambar Lenna hasil penyisipan watermark dan dibandingkan dengan gambar watermark yang telah dimodifikasi. Program kemudian dapat menunjukkan bahwa kedua gambar tersebut berbeda dan mengeluarkan teks yang menunjukkan bahwa gambar watermark yang digunakan untuk melakukan verifikasi tidak tepat dengan watermark yang terdapat pada gambar Lenna hasil penyisipan watermark.

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]] [[0 0 0 ... 6 7 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 7 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]] [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
Wrong watermark!
```

V. ANALISIS

Fragile image watermarking sangat sensitif terhadap perubahan apapun yang dilakukan pada citra. Perubahan-perubahan pada citra jika ditampilkan akan menunjukkan perbedaan dengan watermark pembanding. Hal ini tentu sesuai dengan tujuan dari fragile image watermarking, yaitu untuk memastikan bahwa suatu gambar tidak dimanipulasi dan dimodifikasi.

Pada kasus uji yang telah dilakukan, perbedaan ini terjadi karena fragile image watermarking membangkitkan watermark dari suatu gambar berdasarkan bit LSB nya. Bit LSB ini akan berubah nilainya ketika suatu perubahan terjadi pada gambar utama. Perubahan sekecil apapun, walaupun mungkin hanya 1 bit, dapat dideteksi oleh program sehingga program dapat menentukan bahwa gambar telah dimodifikasi.

Pada pengujian pertama, dilakukan pembangkitan watermark secara normal. Tidak terjadi perubahan apapun dan dapat dilihat bahwa watermark berhasil dibangkitkan dengan baik.

Pada pengujian yang kedua, dilakukan penambahan teks. Penambahan teks rentan terjadi sebagai salah satu tindakan manipulasi citra. Contohnya adalah mengubah nilai nominal pada rekening bank. Penambahan teks dapat merusak watermark yang telah disisipkan pada gambar.

Pada pengujian ketiga, dilakukan penambahan gambar. Penambahan gambar juga sering terjadi dan dapat menimbulkan keraguan keaslian suatu gambar. Dengan adanya fragile image watermarking, dapat dibuktikan keaslian pada suatu gambar apakah terdapat citra lain yang ditambahkan atau tidak.

Pada pengujian keempat, dilakukan pengubahan warna dan kontras. Pengubahan ini mudah dilakukan oleh orang dan seringkali menimbulkan keraguan apakah gambar asli memiliki warna dan kontras yang sama atau tidak. Fragile image watermarking berhasil membuktikan apakah suatu gambar telah diubah warna dan kontrasnya atau tidak.

Pada pengujian kelima, dilakukan pengubahan tipe file. Pengubahan ini diperlukan untuk memastikan bahwa file gambar yang dibuat oleh seseorang sama atau tidak dengan file gambar yang diterima oleh seseorang.

Pada pengujian keenam dan ketujuh, dilakukan pengujian keaslian watermark. Keaslian watermark pada makalah ini juga penting. Jika watermark yang digunakan untuk melakukan verifikasi berubah sedikit saja, program pasti akan menemukan bahwa watermark tersebut telah diubah. Hal ini disebabkan adanya algoritma Blum Blum Shub yang digunakan pada kerandoman watermark. Algoritma Blum Blum Shub menggunakan masukan berdasarkan bit LSB pada gambar watermark. Jika nilai bit LSB tersebut berubah, hasil iterasi m yang didapatkan dari algoritma Blum Blum Shub turut berubah dan hal ini menyebabkan jumlah iterasi pada Arnold's Cat Map ikut berubah dan menyebabkan gambar watermark random yang dihasilkan acak.

Kesimpulan

Dari pembuatan program dan pengujian yang telah dilakukan, penulis menyimpulkan bahwa fragile image watermarking dapat digunakan untuk melindungi dan mengautentikasi suatu gambar. Agar menjadi lebih aman, digunakan Arnold's Cat Map dan Blum Blum Shub guna melakukan random pada gambar watermark sehingga peretas akan menjadi kesulitan dalam melakukan manipulasi gambar.

Ucapan Terima Kasih

Penulis ingin mengungkapkan rasa terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T yang merupakan dosen pengampu mata kuliah IF4020 Kriptografi Semester II Tahun 2022/2023. Selain itu, penulis juga ingin menyampaikan terima kasih kepada rekan-rekan sesama peserta IF4020 Kriptografi yang telah memberikan semangat dan bantuan yang berarti selama proses penulisan makalah ini.

Referensi

[1] Blum, Lenore; Blum, Manuel; Shub, Michael (1983). "Comparison of Two Pseudo-Random Number Generators". Advances in Cryptology. Boston, MA: Springer US. pp. 61–78. doi:10.1007/978-1-4757-0602-4_6. ISBN 978-1-4757-0604-8.

[2] Dyson, Freeman John; Falk, Harold (1992). "Period of a Discrete Cat Mapping". The American Mathematical Monthly. Mathematical Association of America. 99 (7): 603–614. doi:10.2307/2324989. ISSN 0002-9890. JSTOR 2324989.

[3] Hill, Chapel (2022). Geneva May Collins Hall: Arnold's Cat Map: An Exposition. America: University of North Carolina

[4] Ganic, E., Eskicioglu, A.M.: Robust DWT-SVD Domain Image Watermarking: Embedding Data in All Frequencies. In: Proceedings of the ACM Multimedia and Security Workshop, pp. 166–174 (2004)

[5] Munir, Rinaldi. (2012). .Algoritma Enkripsi Citra Digital Berbasis Chaos dengan Penggabungan Teknik Permutasi dan Teknik Substitusi Menggunakan Arnold Cat Map dan Logistic Map Indonesia: Institut Teknologi Bandung.

[6] Vladimir I. Arnold; A. Avez (1967). Problèmes Ergodiques de la Mécanique Classique (in French). Paris: Gauthier-Villars.; English translation: V. I. Arnold; A. Avez (1968). Ergodic Problems in Classical Mechanics. New York: Benjamin

[7] Wei-bin, C., Xin, Z. (2009): Image Encryption Algorithm Based on Henon Chaotic System, Proceeding of International Conference on Image Analysis and Signal Processing (IASP 2009).

[8] Weisstein, Eric W. "Arnold's Cat Map." From MathWorld--A Wolfram Web Resource. https://mathworld.wolfram.com/ArnoldsCatMap.html

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 21 Mei 2023

Marcellus Michael Herman Kahari