APLIKASI VALIDASI CITRA DOKUMEN MENGGUNAKAN *CHAOS* DAN STEGANOGRAFI

Riyandika Andhi Saputra, Suhartono, Priyo Sidik Sasongko

Jurusan Ilmu Komputer/Informatika, FSM, Universitas Diponegoro
riyandika.andhi@gmail.com, suhartono.ilkom@undip.ac.id, priyoss@undip.ac.id

ABSTRAK

Perkembangan teknologi yang pesat memberikan pengaruh pada pengiriman dokumen dari yang semula secara fisik menjadi dalam format citra digital. Namun citra digital tersebut dapat dengan mudah dimodifikasi atau dimanipulasi, serta belum tersedianya aplikasi yang dapat memastikan keaslian citra tersebut. *Chaos* memiliki sifat yang berharga untuk keamanan data, yaitu kepekaan terhadap perubahan kecil pada kondisi awal sistem. Steganografi digunakan untuk menyembunyikan keberadaan tanda keaslian citra dokumen. Dengan menggabungkan kedua metode di atas dibuatlah sebuah aplikasi yang dapat memberikan tanda dan membaca tanda tersebut untuk mengetahui keaslian citra dokumen. Beberapa percobaan yang dilakukan pada aplikasi validasi citra menggunakan *Chaos* dan steganografi berhasil mengenali citra asli dan citra yang sudah dimanipulasi. Didapatkan juga bahwa aplikasi mampu menghasilkan citra dengan PSNR terendah 82,615 dB.

Kata kunci : *Chaos*, steganografi, validasi citra dokumen

I. PENDAHULUAN

Citra digital memiliki banyak kelebihan dibandingkan berbentuk fisik, salah satunya yaitu mudah diduplikasi dan dikirimkan dengan waktu yang singkat. Hal ini kemudian dimanfaatkan untuk pengiriman dokumen berbentuk citra digital sebagai keperluan registrasi pengajuan seperti beasiswa *online* berupa dokumen ijazah atau sertifikat.

Namun sayangnya citra digital tersebut dapat dengan mudah dimodifikasi atau dimanipulasi. Di sisi lain, belum ada layanan yang tersedia untuk memastikan dokumen tersebut asli atau sudah dimanipulasi. Manipulasi tersebut dapat berupa penggantian nama, memalsukan tanda tangan, atau mengubah nilai huruf pada transkrip nilai. Oleh karena itu diperlukan aplikasi validasi citra yang sensitif terhadap perubahan kecil sekalipun. Agar memiliki sifat sensitif dipilihlah *Chaos* karena memiliki sifat peka terhadap nilai awal sebagai pembangkit kunci acak pada proses penyandian tanda citra asli.

Setelah tanda tersandikan, diperlukan suatu cara untuk menyampaikannya beserta dengan gambar yang akan divalidasi keasliannya, untuk itu diperlukan steganografi yang merupakan suatu ilmu yang mempelajari tentang penyembunyian pesan di dalam suatu media yang disebut *carrier* atau *cover object* sehingga keberadaan pesan tersebut tersembunyi [11].

Proses memastikan keaslian dari citra digital dokumen dilakukan dengan masukan berupa citra digital yang telah diolah menggunakan *Chaos* dan steganografi. Pesan yang berupa sandi dalam citra tersebut diambil. Kemudian dengan menggunakan *Chaos* sebagai pembangkit kunci acak pesan tersebut diubah menjadi tanda untuk mengenali citra tersebut asli atau sudah dimanipulasi.

II. KRIPTOGRAFI

Kriptografi adalah ilmu penulisan rahasia dengan tujuan menyembunyikan arti dari pesan [6]. Kriptografi terdiri dari dua buah proses, yaitu enkripsi dan dekripsi yang keduanya membutuhkan kunci untuk prosesnya.

Suatu algoritma dikatakan aman, apabila belum ada cara untuk menemukan *plaintext*-nya. Sampai saat ini, hanya algoritma *One Time Pad* (OTP) yang dinyatakan sebagai *unbreakable cipher* karena panjang kunci yang berupa barisan bilangan acak sama dengan panjang *plaintext* [6].

Jika karakter yang digunakan adalah anggota himpunan 256 karakter (seperti karakter dengan pengkodean ASCII), maka enkripsi dapat dinyatakan sebagai penjumlahan modulo 256 dari satu karakter plainteks dengan satu karakter kunci *one time pad*:

$$c_i = (p_i + k_i) \bmod 256 \qquad (1)$$

Penerima pesan menggunakan pad yang sama untuk mendekripsikan cipherteks menjadi plainteks dengan persamaan:

$$p_i = (c_i + k_i) \bmod 256 \qquad (2)$$

Fungsi *hash* adalah fungsi yang menerima masukan *string* yang panjangnya sembarang, lalu mentransformasikannya menjadi string keluaran yang panjangnya tetap (*fixed*). Penggunaan fungsi hash biasanya terdapat pada tanda tangan digital dan otentikasi pesan [9].

III. *CHAOS*

Teori *Chaos* berasal dari teori sistem yang memperlihatkan kemunculan yang tidak teratur, meskipun sebenarnya teori ini digunakan untuk menjelaskan kemunculan data acak [6].

Fenomena yang umum di dalam teori *Chaos* adalah peka terhadap perubahan nilai awal, yang juga dikenal sebagai ketergantungan yang peka pada nilai awal (*sensitive dependence on initial condition*).

Salah satu fungsi *Chaos* sederhana adalah persamaan logistik di dalam ekologi yang digunakan untuk mensimulasikan pertumbuhan populasi spesies [6], yaitu

$$f(x) = r\,x\,(1 - x) \qquad (3)$$

Fungsi ini dapat dinyatakan dalam bentuk iteratif.

$$x_{i+1} = r\,x_i\,(1 - x_i) \qquad (4)$$

Di dalam persamaan (3) dan (4) di atas x adalah populasi spesies pada interval waktu yang ditentukan dengan x_0 adalah nilai awal iterasi. Daerah asal x adalah dari 0 sampai 1, yang dalam hal ini 1 menyatakan populasi maksimum dan 0 yang menyatakan kepunahan, sedangkan $0 \leq r \leq 4$. Konstanta r menyatakan laju pertumbuhan.

IV. STEGANOGRAFI

Steganografi berasal dari bahasa Yunani "Steganos" yang berarti tulisan tersembunyi. Dengan kata lain, steganografi adalah ilmu dan seni menyembunyikan informasi dengan cara menyisipkan (*embedding*) pesan di dalam pesan lain [9].

Metode steganografi dapat diklasifikasikan menjadi *spatial domain embedding* dan *frequency domain embedding*. Frequency domain embedding penyisipan pesan dilakukan pada koefisien transformasi. *Spatial domain embedding* bekerja dengan memodifikasi langsung nilai *byte* dari *cover-object* yang merepresentasikan intensitas atau warna *pixel*. Agar metode *spatial domain embedding* ini tidak terlihat secara kasat mata maka *bit* yang dirubah adalah *bit* paling akhir atau biasa disebut *least significant bit* (LSB) karena kemudahannya maka metode ini dipilih untuk aplikasi validasi citra dokumen menggunakan *Chaos* dan steganografi.

V. APLIKASI VALIDASI CITRA DOKUMEN MENGGUNAKAN *CHAOS* DAN STEGANOGRAFI

Kemudahan penggunaan format digital diiringi dengan semakin mudahnya untuk dimodifikasi maupun dimanipulasi, sehingga tidak jelas lagi mana dokumen yang asli dengan yang sudah mengalami perubahan. Untuk itu aplikasi ini dibuat dengan harapan mampu mengenali mana citra dokumen yang asli atau tidak.

Aplikasi validasi citra dokumen menggunakan *chaos* dan steganografi terbagi menjadi dua proses, yaitu enkoding dan dekoding. Proses enkoding dilakukan oleh pengurus aplikasi yaitu dengan memasukkan citra yang akan diberi tanda berupa hash citra dan teks. Secara umum proses enkoding terdiri dari:

a. Pengaturan aplikasi

Bagian ini berfungsi untuk mengambil pengaturan aplikasi yang telah disimpan dalam basis data. Pengaturan ini digunakan sebagai parameter aplikasi. Pengaturan yang terdapat pada basis data:

1) Nilai r : digunakan untuk parameter pada *chaos*.
2) Jumlah iterasi : digunakan untuk parameter pada chaos, yaitu berapa banyak iterasi yang dibutuhkan dalam pembangkitan kunci.
3) Algoritma hash : untuk menentukan algoritma hash apa yang digunakan.
4) Kunci aplikasi : untuk mencegah duplikasi aplikasi, dan membedakan aplikasi satu dengan lainnya.

b. Pembuatan hash citra

Proses pembuatan hash citra diawali dengan membuat string citra, yaitu:

```
FOR jumlah kolom Citra DO
  FOR jumlah baris Citra DO
      Ambil nilai RGB
      Ubah nilai RGB menjadi genap
      Stream ← Stream + konversi nilai
          RGB menjadi karakter
```

Pada algoritma di atas string citra disusun oleh nilai RGB yang diubah menjadi genap karena pada steganografi LSB bekerja pada bit paling akhir, sehingga string citra tidak menyertakan bit paling akhir. Setelah string citra terbentuk kemudian diubah menjadi hash citra sesuai dengan algoritma yang dipilih.

c. Menyusun hash dan teks

Hash citra yang sudah diperoleh tadi kemudian disusun sebagai *string* dengan teks. Untuk mengetahui batas antara hash citra dan teks, di-*generate* suatu *boundary* yang membatasinya. Sehingga akan tercipta suatu string yang terdiri dari:

boundary + hash citra + *boundary* + teks + *boundary*

d. Ubah hash menjadi kunci awal

Proses ini berfungsi untuk menyiapkan kunci awal bagi *chaos* yang berupa nilai antara 0 dan 1. Kunci ini berasal dari hash citra yang tadi sudah dibuat dan dikombinasikan dengan hash kunci yang terdapat pada pengaturan aplikasi.

Masing-masing hash dipotong menjadi 7 karakter dikarenakan batasan dari aplikasi yang hanya bekerja dalam 32-bit. Pada bagian akhir algoritma baru kemudian diubah menjadi nilai desimal.

```
Potong1 ← ubah HashCitra menjadi array
    string dengan panjang 7 karakter per
    elemennya
Potong2 ← ubah HashKunci menjadi
    array string dengan panjang 7 karakter
    per elemennya
Hasil ← 0
FOR i←0 TO Jumlah elemen Potong1 DO
    Irisan ← nilai integer Potong1 XOR
    nilai integer Potong2
    IF i MOD 3 = 0 THEN
        Temp ← Irisan
        IF (i+1) = Jumlah elemen Potong1
        THEN
            Hasil ← Hasil XOR Temp
    ELSE IF i MOD 3 = 1
        Temp ← Temp + Irisan
        IF (i+1) = Jumlah elemen Potong1
        THEN
            Hasil ← Hasil XOR Temp
    ELSE
        Hasil ← Hasil XOR Temp
Hasil ← Hasil / (10 Pangkat 9)
    → Hasil
```

e. Enkripsi OTP dengan *chaos* sebagai pembangkit kunci acak

String *plaintext* yang didapat dari susunan hash dan teks dienkripsi menggunakan OTP dengan menggunakan *chaos* dan menghasilkan *ciphertext*.

f. Steganografi LSB ke dalam citra

Bagian akhir pada proses dekoding, yaitu menyisipkan *ciphertext* ke dalam citra dengan menggunakan steganografi LSB.

Sebaliknya pada proses dekoding, dilakukan oleh pengunjung aplikasi untuk memastikan keaslian citra yang dimilikinya dengan memasukkan citra yang akan diperiksa keasliannya. Proses dekoding ini terdiri dari beberapa bagian, yaitu:

a. Pengaturan aplikasi
b. Pembuatan hash citra
c. Ubah hash menjadi kunci awal

d. Ambil *ciphertext*, dekripsi OTP dengan *chaos*, dan perbandingan hash

Pada bagian awal proses dekoding yaitu poin a hingga c sama dengan yang terdapat pada proses enkoding, yaitu untuk mendapatkan pengaturan awal proses.

Pada poin d, terdiri dari beberapa fungsi, yaitu:

1) Pengambilan *ciphertext* yang disisipkan menggunakan steganografi
2) Dekripsi OTP dan pembangkitan kunci menggunakan *chaos*
3) Perbandingan hash

Ketiga proses tersebut berjalan berulang-ulang dalam satu kesatuan blok iterasi. Hal ini bertujuan untuk menghemat penggunaan memori, sehingga aplikasi tidak perlu membaca keseluruhan *pixel* citra hingga akhir, namun cukup hingga ditemukan *boundary* ketiga atau ketika terjadi ketidakcocokan hash citra dengan hash pada plainteks maka iterasi dapat diakhiri.

VI. PEMBAHASAN

Pengujian dilakukan dengan pengaturan aplikasi: Nilai r = 3.99, jumlah iterasi = 120, algoritma hash = md5, dan kunci aplikasi = "selamat". Gambar yang digunakan berupa hasil scan citra dokumen. Dari pengaturan di atas diperoleh hasil seperti pada Gambar 1.

Pengujian dilakukan dengan menggunakan citra asli pada Gambar 1 dan yang telah dimanipulasi pada Gambar 2. Pengukuran PSNR menggunakan rumus

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[f'(i,j) - f(i,j)\right]^2$$

$$PSNR = 10\ \log_{10}\left[\frac{255\ \times 255}{MSE}\right] dB$$

Hasil dari pengujian ini terdapat pada Tabel 1.

No	Citra	Keterangan	Hasil	PSNR
1	Gambar 1a	Citra ijazah	Asli	85.928
2	Gambar 1b	Citra halaman pengesahan	Asli	82.615
3	Gambar 1c	Citra transkrip nilai	Asli	85.899
4	Gambar 2a	Citra ijazah, manipulasi nama	Tidak asli	-
5	Gambar 2b	Citra ijazah, manipulasi tanda tangan dengan penambahan titik	Tidak asli	-
6	Gambar 2c	Citra halaman pengesahan, manipulasi nama	Tidak asli	-
7	Gambar 2d	Citra halaman pengesahan, manipulasi tanda tangan dengan penambahan titik	Tidak asli	-
8	Gambar 2e	Citra transkrip nilai, manipulasi nama	Tidak asli	-
9	Gambar 2f	Citra transkrip nilai, manipulasi nilai B menjadi A	Tidak asli	-
10	Gambar 2g	Citra transkrip nilai, manipulasi IPK 3,61 menjadi 3,81	Tidak asli	

a b c

Gambar 1 Citra asli

a

b

c

d

e

F

g

Gambar 2 Citra Manipulasi

VII. KESIMPULAN

Aplikasi validasi citra dokumen menggunakan *chaos* dan steganografi merupakan aplikasi berbasis *web* yang dapat memastikan keaslian dari citra dokumen. Dari hasil pengujian nampak bahwa aplikasi mampu mengenali citra asli dan yang sudah dimanipulasi. Berdasarkan hasil pengujian, citra dokumen yang dihasilkan memiliki PSNR terendah sebesar 82.615dB, hal ini menunjukkan bahwa citra yang dihasilkan tidak nampak perbedaannya secara kasat mata. Sedangkan dari 10 pengujian berbagai citra, aplikasi mengenali semuanya dengan tepat. Bahkan aplikasi dapat mengenali perubahan kecil berupa penambahan titik seperti pada pengujian 7 dan 8.

DAFTAR PUSTAKA

[1] Arlow, Jim, et all., 2002, "*UML and The Unified Process"*, Boston : Addison Wesley.

[2] Booch, Grady, et all., 2005, "*The Unified Modelling Language User Guide Second Edition*", Boston : Addison Wesley.

[3] Chen C., 1998, "*On The Selection of Image Compression Algorithms*", NSC Grant Departement of Computer Science National Tsing Hua University, Taiwan.

[4] Fowler, Martin, 2003, "*UML Distilled: A Brief Guide to the Standard Object Modelling Language, Third Edition*", Boston : Addison Wesley.

[5] Jacobson, Ivar, et all., 1999, "*The Unified Software Development Process*", Boston : Addison Wesley.

[6] Munir R., Riyanto B., dan Sutikno S., 2006, "*Perancangan Algoritma Kriptografi Stream Cipher dengan Chaos*", Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[7] Paar C. dan Pelzl J, 2010, "*Understanding Cryptography A Textbook for Students and Practitioners*" , Springer, New York.

[8] Pressman, Roger S, 2001, "*Software Engineering : A Practitioner's Approach Fifth Edition*", New York , McGraw - Hill Companies, Inc.

[9] Silman, Joshua, 2001, "*Steganography and steganalysis: an overview*", Tech. Rep., SANS Institute.

[10] Stallings, William, 2005, "*Cryptography and Network Security Principles and Practices, Fourth Edition*",Prentice Hall.

[11] Yu L., Zhao Y., Ni R., dan Li T., 2010, "*Improved Adaptive LSB Steganography Based on Chaos and Genetic Algorithm*", Hindawi Publishing Corporation, EURASIP Journal on Advances in Signal Processing.