Jurnal Coding, Sistem Komputer Untan ISSN: 2338-493X
Volume 04. No. 2 (2016), hal 119 -129

PENERAPAN KRIPTOGRAFI MENGGUNAKAN ALGORITMA KNAPSACK, ALGORITMA GENETIKA, DAN ALGORITMA ARNOLD'S CATMAP PADA CITRA

[1]Martinus Dias, [2]Cucu Suhery, [3]Tedy Rismawan
[1][2][3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail:
[1]martinusdias0@student.untan.ac.id, [2]csuhery@siskom.untan.ac.id,
[3]tedyrismawan@siskom.untan.ac.id

Abstrak

Kriptografi tidak hanya dapat diterapkan pada data tulisan/teks, tetapi dapat juga diterapkan pada data yang berupa citra/gambar. Penelitian ini menerapkan kriptografi pada data yang berupa citra. Proses enkripsi dilakukan dua kali, yang pertama adalah untuk pembentukan kunci dengan menggunakan Algoritma Genetika dan Algoritma Knapsack. Kunci yang dihasilkan menjadi masukan dalam proses enkripsi yang kedua. Proses enkripsi yang kedua bertujuan untuk mengacak citra dengan menggunakan algoritma Arnold's Catmap. Proses dekripsi juga dilakukan dua kali, dimana proses yang pertama adalah membentuk kunci dengan menggunakan Algoritma Genetika dan Algoritma Knapsack, yang akan menghasilkan kunci yang akan digunakan pada proses dekripsi. Proses dekripsi menggunakan Algoritma Arnold's Catmap sehingga dihasilkan citra semula. Citra yang digunakan pada penelitian ini berekstensi png, jpeg, dan bmp. Hasil enkripsi dan dekripsi dapat disimpan dengan ekstensi citra png, jpeg, atau bmp sesuai keinginan dari pengguna aplikasi. Hasil pengujian kriptografi yang diterapkan pada citra berekstensi bmp dan png berhasil dengan baik, yaitu citra hasil dekripsi terlihat seperti citra semula. Namun pengujian terhadap citra yang berekstensi jpeg, hasil dekripsinya terdapat noise.

Kata kunci: Algoritma Knapsack, Algoritma Genetika, Algoritma Arnold's Catmap, kriptografi citra

1. PENDAHULUAN

Pencurian data sering terjadi dalam dunia maya atau yang lebih kita kenal dunia internet. Berbagai data dapat dicuri dengan mudah dalam internet seperti data pribadi, data keuangan, dan berbagai jenis data-data lainnya. Salah satu data yang dapat dicuri adalah data citra/gambar. Bagaimana jika data yang dicuri adalah sebuah desain bangunan atau desain pakaian yang belum dipublikasikan, citra-citra tersebut bisa saja diklaim oleh pihak-pihak tertentu sehingga dapat merugikan pihak yang membuat karya seni tersebut. Untuk itu diperlukan suatu keamanan khusus untuk melindungi desain-desain yang belum dipublikasikan tersebut.

Citra merupakan salah satu bentuk data atau informasi yang disajikan secara visual [1]. Citra sering digunakan sebagai media untuk menampilkan seni, foto satelit, medis dan sebagainya. Untuk melindungi data citra, maka perlu sistem keamanan data yaitu kriptografi. Dengan kriptografi, data dapat dikodekan dengan algoritma tertentu sehingga orang yang tidak berkepentingan tidak dapat mengakses data tersebut.

Salah satu teknik kriptografi adalah metode algoritma knapsack dan algoritma genetika. Pada penelitian yang dilakukan Rahmi [2], algoritma knapsack dan algoritma genetika dapat digunakan bersama pada media tulisan. Pada penelitian ini, penelitian Rahmi [2] dikembangkan dengan menerapkan algoritma knapsack dan algoritma genetika ke dalam media citra.

 ISSN: 2338-493X

Berdasarkan masalah yang dipaparkan, maka dilakukan penelitian dengan judul "Penerapan Kriptografi Algoritma Knapsack, Algoritma Genetika dan Algoritma Arnold's Catmap Pada Citra". Pengembangan yang dilakukan pada penelitian ini adalah menghilangkan deret *superincreasing* dan menerapkan Algoritma Knapsack dan Algoritma Genetika pada kriptografi citra dengan Algoritma Arnold's Catmap.

2. TINJAUAN PUSTAKA

2.1. Citra Digital

Suatu citra dapat didefinisikan sebagai fungsi dua dimensi, f (x,y) dimana x (baris) dan y (kolom) merupakan koordinat pada bidang datar yang berhubungan. Citra digital dibentuk oleh kumpulan titik yang dinamakan *pixel*. *Pixel* merupakan titik pertemuan antara sumbu x dan sumbu y gambar [3].

Tiga jenis citra yang sering digunakan yaitu citra berwarna, citra berskala keabuan, dan citra biner [3].

a. Citra warna

Citra warna atau citra RGB adalah citra yang menyajikan warna dalam bentuk tiga komponen yaitu R (merah), G (hijau), dan B (biru). Setiap komponen warna terdiri dari 8 bit dengan nilai berkisar antara 0-255. Dengan demikian, kemungkinan warna yang disajikan 255 x 255 x 255 warna.

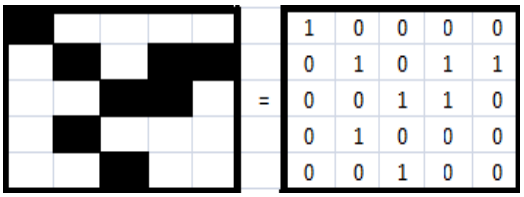
b. Citra Berskala Keabuan

Citra jenis ini memiliki warna yang lebih sederhana yaitu hanya memiliki warna hitam dan putih. Pada citra berskala keabuan, warna dinyatakan dengan intensitas. Intensitas gambar berkisar antara 0-255. nilai 0 menyatakan hitam dan nilai 255 menyatakan putih seperti gambar 1.

Gambar 1. Citra Berskala Keabuan.

c. Citra Biner

Citra biner merupakan citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua kemungkinan yaitu 0 dan 1. Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan putih seperti gambar 2.



Gambar 2. Citra Biner.

2.2. Algoritma Genetika

Algoritma Genetika adalah metode pencarian yang didasarkan pada proses evolusi alamiah, yaitu terbentuknya populasi awal secara acak yang terdiri dari individu-individu dengan sifat yang tergantung pada gen-gen dalam kromosomnya. Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik ke dalam satu kumpulan, sehingga dihasilkan generasi terbaik. Dalam evolusi alam, individu yang bernilai *fitness* tinggi akan bertahan hidup sedangkan yang bernilai *fitness* rendah akan mati [4].

Algoritma Genetika merupakan proses pencarian yang *heuristic* dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetika dalam menemukan solusi optimum suatu masalah. Hal yang harus diperhatikan adalah menghindari terjadinya *konvergensi premature*, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal.

Operator genetika yang digunakan setelah proses evaluasi tahap pertama membentuk populasi baru dari generasi sekarang. Operator-operator tersebut adalah operator seleksi, *crossover* dan mutasi [4].

a. *Selection* (seleksi)

Seleksi adalah proses penentuan individu mana yang akan menjadi *parent* di generasi berikutnya. Secara alamiah, individu yang lemah tidak akan bertahan

 ISSN: 2338-493X

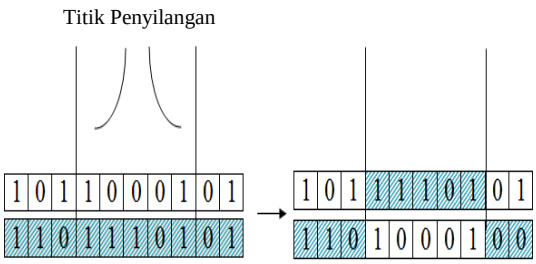
hidup lama untuk bereproduksi. Dengan kata lain, probabilitas individu diturunkan, proporsional dengan nilai *fittness*-nya [4].

b. *Crossover*

Crossover (perkawinan silang) bertujuan menambah keanekaragaman kromosom. Dengan penyilangan antar kromosom satu dengan kromosom lainnya akan menghasilkan kromosom berbeda dari kromosom induk. Beberapa jenis *crossover* tersebut adalah [4]:

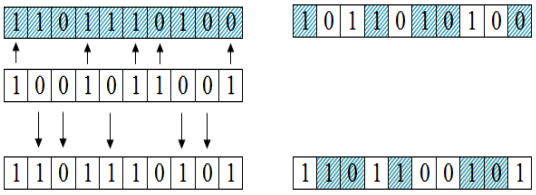
b1. Penyilangan yang Melibatkan Kode Biner.

- Penyilangan N-titik (*N-Point Crossover*)
  Pada metode ini setiap kromosom induk dipotong menjadi N+1 bagian. Kromosom anak pertama akan mewariskan bagian potongan ganjil induk yang pertama dan potongan genap dari induk yang kedua. Demikian juga dengan kromosom anak kedua dihasilkan dari dari sisa kromosom induknya seperti Gambar 3.



Gambar 3. Metode penyilangan n-titik.
Sumber: Zukhri (2014).

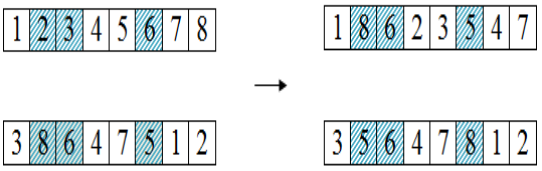
- Penyilangan Seragam (*Uniform Crossover*)
  Penyilangan seragam memerlukan pembangkitan topeng penyilangan yang terdiri dari kode biner sebanyak gen dalam kromosom. Jika kode pada kedudukan topeng penyilangan bernilai 1 maka yang diwariskan berasal dari gen induk pertama, dan jika kode pada topeng penyilangan bernilai 0 maka gen yang diwariskan berasal dari induk kedua. Gen induk yang diwariskan berdasarkan topeng penyilangan kemudian dikelompokan dan akan menjadi anak 1 sedangkan gen-gen yang tidak diwariskan akan dikelompokan kembali menjadi anak kedua. Untuk memperjelas metode penyilangan dapat dilihat pada Gambar 4.



Gambar 4. Metode penyilangan seragam.
Sumber: Zukhri (2014).

b2. Penyilangan untuk Optimasi Kombinatorial.

- Penyilangan Berbasis Posisi (*Position Based Crossover*)
  Penyilangan berbasis posisi dilakukan dengan memilih sejumlah posisi gen secara acak. Gen-gen pada posisi terpilih pada induk yang pertama diwariskan pada kromosom anak yang kedua, sedangkan gen-gen terpilih dari kromosom anak yang kedua akan diwariskan pada anak pertama dengan urutan yang sama. Ilustrasi metode penyilangan seperti terlihat pada Gambar 5.

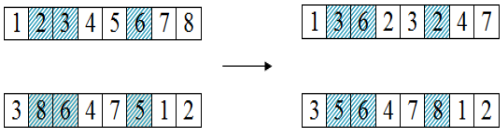


Gambar 5. Metode penyilangan berbasis posisi.
Sumber: Zukhri (2014).

- Penyilangan Berbasis Urutan (*Order-Based Crossover*)

Pada metode ini, dipilih posisi gen-gen secara acak, kemudian dibentuk kromosom anak yang pertama pada posisi gen-gen terpilih tersebut dengan cara mengambil dari gen-gen pada posisi terpilih dari induk yang pertama tetapi dengan urutan mengikuti gen-gen dengan nilai yang sama dari induk yang kedua seperti pada Gambar 6.

 ISSN: 2338-493X



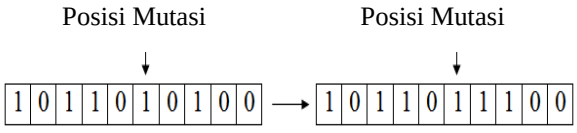
Gambar 6. Metode penyilangan berbasis urutan**.**
Sumber: Zukhri (2014).

c. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Untuk semua gen yang ada, jika bilangan random yang dibang-kitkan kurang dari probabilitas mutasi P_{mut} yang ditentukan maka ubah gen tersebut menjadi nilai kebalikannya jika bernilai 0 diubah menjadi 1 dan jika nilai 1 diubah menjadi 0 [5].

Operator mutasi merupakan operasi yang menyangkut satu kromosom tertentu. Beberapa cara operasi mutasi diterapkan dalam algoritma genetika menurut jenis pengkodean terhadap *phenotype*, salah satu jenis mutasi adalah mutasi yang melibatkan kode biner[4]:

Pada mutasi yang melibatkan kode Biner ini, merepresentasikan kromosom dengan kode biner, operasi mutasi dapat dilakukan dengan mengubah nilai gen pada posisi tertentu. Gen yang akan mengalami mutasi dipilih secara acak. Bentuk mutasi yang terjadi adalah mengubah nilai gen yang semula satu menjadi nol, atau nol berubah menjadi satu. Contoh mutasi dapat terlihat pada Gambar 7.



Gambar 7. Metode mutasi untuk kode biner.
Sumber: Zukhri (2014).

2.2 Knapsack

Knapsack dapat diartikan sebagai sebuah karung yang dapat digunakan untuk memasukan sejumlah barang. Karung tersebut memiliki kapasitas yang terbatas, sehingga tidak semua barang dapat masuk ke dalam karung. Untuk mengoptimalkan penggunaan kapasitas karung yang terbatas, maka perlu pemilihan yang tepat terhadap jenis barang yang dimasukkan. Optimasi pada penelitian ini dapat dihitung dengan menggunakan persamaan (1) [6]:

$$M = b_1 w_1 + b_2 w_2 + \cdots + b_n w_n \qquad (1)$$

Ket:

M = Problem Knapsack
b_n = biner plainteks
w_n = himpunan kunci
n = banyak deret Knapsack

Nilai M merupakan problem Knapsack, yang didapat dari penjumlahan perkalian biner plainteks (b_n) dan himpunan kunci (w_n). Biner plainteks pada penelitian ini menggunakan biner yang dihasilkan oleh proses mutasi pada Algoritma Genetika. Himpunan kunci merupakan kunci masukan yang diubah ke dalam bentuk desimal berdasarkan kode ASCII.

2.3. Arnold's Catmap

Algoritma Arnold's Cat Map dapat didefenisikan dengan persamaan (2) :

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} mod\ (N) \qquad (2)$$

Pada persamaan (2) $\begin{bmatrix} X_i \\ Y_i \end{bmatrix}$ adalah posisi piksel (x,y) di dalam citra berukuran NxN dan $\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix}$ adalah posisi piksel (X_{i+1}, Y_{i+1}) posisi piksel yang baru setelah transfor-masi, b dan c adalah bilangan bulat positif sembarang [7]. Matriks $\begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}$ meru-pakan matriks determinan yang digu-nakan untuk mengiterasi posisi piksel.

Untuk proses dekripsi pada persamaan arnold's cat map adalah kebalikan dari proses enkripsinya yaitu sesuai dengan persamaan (3) [7]:

$$\begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} mod\ (N) \qquad (3)$$

3. METODOLOGI PENELITIAN

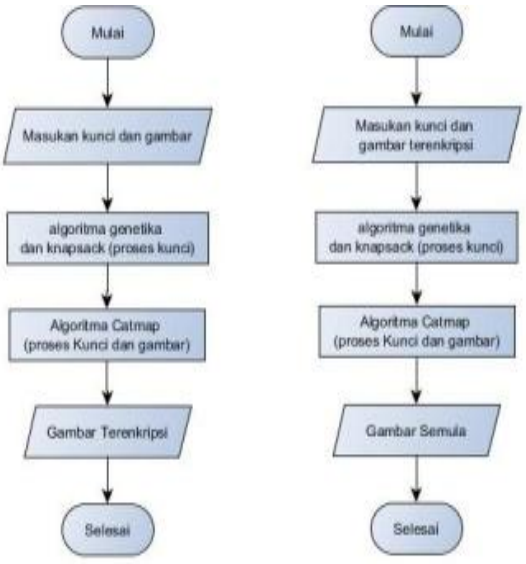
Penelitian ini menggunakan meto-dologi yang dimulai dari studi pustaka tentang algoritma yang digunakan. Selanjutnya akan dilakukan analisis

 ISSN: 2338-493X

kebutuhan. Analisis kebutuhan terdiri dari analisis kebutuhan perangkat keras dan perangkat lunak yang digunakan. Setelah kebutuhan diketahui, maka akan dilakukan perancangan sistem. Dalam perancangan sistem terdapat perancangan DFD, perancangan antar muka, perancangan *flowchart* kriptografi. Proses terakhir pada metodologi penelitian ini adalah pengujian. Pengujian yang dilakukan adalah pengujian sensitivitas kunci, pengujian kriptografi, serta pengujian *black box*.

4. PERANCANGAN SISTEM

Perancangan sistem yang dibuat seperti terlihat pada Gambar 8. Sistem dibuat dengan dua fungsi yaitu enkripsi dan dekripsi citra. Apabila pengguna ingin mengenkripsi citra, maka pengguna diminta untuk mengisi kunci dan citra yang akan dienkripsi. Kunci masukan kemudian melalui proses kriptografi dengan menggunakan algoritma knapsack dan algoritma genetika. Hasil enkripsi dari kunci kemudian digunakan untuk mengenkripsi citra dengan menggunakan algoritma *arnold's cat map*. Hasil yang didapat dari proses enkripsi adalah citra terenkripsi.



(a) (b)

Gambar 8. Perancangan Aplikasi Kriptografi. (a) enkripsi (b) dekripsi

Setelah didapat citra terenkripsi, citra tersebut dapat didekripsi atau jika tidak citra akan selesai diproses. Apabila citra didekripsi pengguna harus memasukkan citra yang telah terenkripsi dan kunci yang sama dengan yang digunakan pada saat enkripsi. Kunci yang menjadi masukan kemudian dienkripsi dengan menggunakan algoritma knapsack dan algoritma genetika. Hasil dari kriptografi kunci kemudian digunakan untuk mendekripsi citra sehingga dihasilkan citra semula yang sama dengan citra sebelum terenkripsi.

4.1. Perancangan Model Kriptografi

Perancangan model kriptografi merupakan perancangan yang berisi penetapan masukan, penetapan keluaran serta menceritakan arsitektur kriptografi yang digunakan. Perancangan model kriptografi akan dijelaskan kriteria-kriteria yang berlaku pada proses masukan maupun keluaran data pada kriptografi.

a. Penetapan Masukan

Penetapan masukan bertujuan untuk mengetahui apa saja yang dapat menjadi masukan pada aplikasi, sehingga tidak terjadi kesalahan pada proses kriptografi. Masukan yang diperlukan pada aplikasi adalah sebagai berikut:

1. Masukan Pada Proses Enkripsi
   Masukan yang digunakan pada proses enkripsi adalah kunci dan citra yang akan dienkripsi. Untuk proses enkripsi pada aplikasi ini, kunci yang diperlukan harus memiliki 1-20 karakter serta citra yang memiliki ekstensi png, jpeg, dan bmp.
2. Masukan Pada Proses Dekripsi
   Masukan untuk melakukan dekripsi adalah kunci yang sama untuk proses enkripsi dan citra yang telah dienkripsi. Pada proses dekripsi masukan yang diperlukan adalah kunci yang sama pada proses enkripsi, serta gambar yang diproses pada dekripsi memiliki ekstensi png, jpeg, dan bmp.

b. Penetapan Keluaran

Penetapan keluaran adalah pembatasan keluaran apa yang dihasilkan oleh sistem

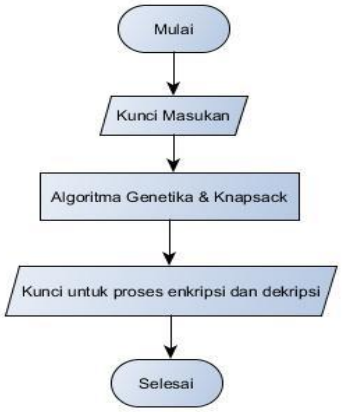
 ISSN: 2338-493X

setelah berproses. Batasan keluaran yang harus dipenuhi adalah sebagai berikut:

1. Keluaran Pada Proses Enkripsi
   Keluaran untuk proses enkripsi adalah citra terenkripsi dengan ekstensi png, jpeg, dan bmp.
2. Keluaran Pada Proses Dekripsi
   Keluaran yang didapat pada proses dekripsi adalah citra seperti citra sebelum dienkripsi/citra semula dengan ekstensi png, jpeg, dan bmp.

4.2. Proses Pembentukan Kunci

Proses pembentukan kunci mempunyai alur yang dapat dilihat pada Gambar 9. Proses ini merupakan proses yang dilalui oleh sebuah kunci sebelum dimplementasikan ke dalam citra.



Gambar 9. *Flowchart* Proses Pembentukan Kunci

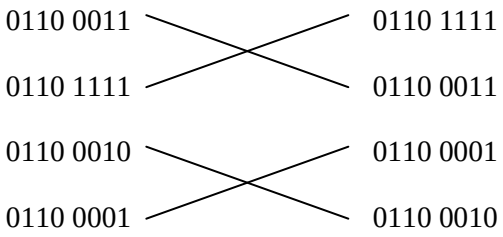
Kunci yang menjadi masukan pada aplikasi akan diproses dengan menggunakan algoritma knapsack dan algoritma genetika. Hasil yang didapat dari proses algoritma knapsack dan algoritma genetika pada kunci awal adalah sebuah kunci baru yang akan menjadi kunci untuk memproses citra baik untuk proses enkripsi maupun dekripsi. Panjang kunci yang dapat dimasukan ke dalam aplikasi ini adalah 1-20 karakter. Untuk memperjelas proses pembentukan kunci dapat dilihat pada contoh sebagai berikut:

Contoh: Kunci yang dimasukkan adalah "coba". Kunci "coba" tersebut kemudian dikonversi ke dalam bentuk desimal ASCII. Dari bentuk desimal, kunci tersebut dikonversikan kembali ke dalam bentuk biner :

coba = 99, 111, 98, 97 → desimal

biner = 01100011 01101111 01100010 01100001

crossover



Sehingga hasil *crossover* yang didapat adalah: 01101111 01100011 01100001 01100010. Hasil *crossover* kemudian akan dimutasi sehingga didapatkan hasil mutasi sebagai berukut:

Crossover : 01101111 01100011 01100001 01100010

Mutasi: 10010000 10011100 10011110 10011101.

Setelah selesai melakukan proses Algoritma Genetika, kunci kemudian akan diproses dengan menggunakan Algoritma Knapsack. Algoritma Knapsack yang dilakukan pada penelitian ini adalah dengan mengalikan hasil mutasi terhadap perulangan kunci yang dikonversi kebentuk desimal. Banyak perulangan kunci tergantung pada banyak biner hasil mutasi. Untuk memperjelas dapat dilihat pada perhitungan berikut:

10010000 10011100 10011110 10011101

99 111 98 97 99 111 98 97 99 111 98 97 99 111 98 97 99 111 98 97 99 111 98 97 99 111 98 97

Sehingga:

Hasil knapsack = (1x99) + (0x111) + (0x98) + (1x97) + (0x99) + (0x111) + (0x98) + (0x97) + (1x99) + (0x111) + (0x98) + (1x97) + (1x99) + (1x111) +

 ISSN: 2338-493X

(0x98) + (0x97) + (1x99) + (0x111) + (0x98) + (1x97) + (1x99) + (1x111) + (1x98) + (0x97) + (1x99) + (0x111) + (0x98) + (1x97) + (1x99) + (1x111) + (0x98) + (1x97)

Hasil knapsack = 1609.

Kunci yang dihasilkan pada proses algoritma genetika dan algoritma knapsack adalah 1609.

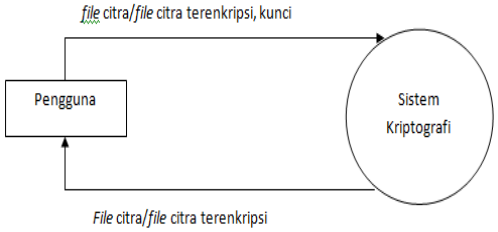
4.3 Proses Catmap

Proses Catmap merupakan proses untuk mengubah citra dari citra asal menjadi citra terenkripsi sehingga bentuk asli dari citra tidak terlihat. Proses Catmap dapat digunakan juga untuk mengembalikan citra terenkripsi menjadi citra semula. Proses Catmap akan menyebabkan citra melakukan iterasi berdasarkan masukan kunci yang dihasilkan dari Algoritma Knapsack dan Algoritma Genetika. Dengan masukan citra 300 x 300 *pixel* maka citra akan beriterasi sebanyak 300 kali untuk kembali kebentuk semula [8]. Agar citra tidak kembali kebentuk semula maka jumlah iterasi yang dilakukan harus kurang dari 300 kali. pada penelitian ini proses iterasi yang dilakukan sebanyak 299 kali, sehingga untuk proses dekripsi diperlukan 1 kali iterasi.

4.4 Perancangan DFD Kriptografi

a. Diagram Konteks

Diagram konteks untuk sistem yang akan dibuat dapat dilihat pada Gambar 10. Sistem yang dibangun memiliki satu terminator yaitu pengguna. Aliran data yang masuk berupa *file* citra, kunci untuk proses enkripsi, serta citra terenkripsi dan kunci untuk proses dekripsi.



Gambar 10. Diagram Konteks

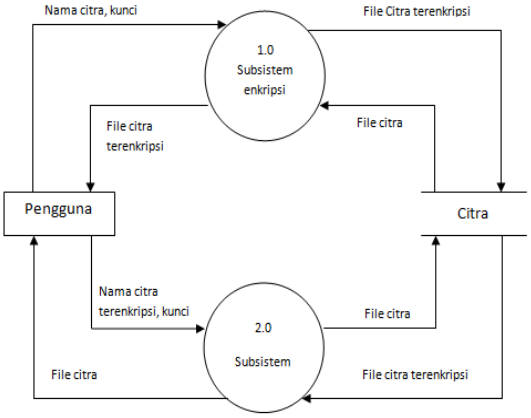
b. Data *Flow* Diagram

DFD Level 0

DFD yang dibuat pada Gambar 11 merupakan subsistem dari sistem kriptografi yang terdapat pada Gambar 10. Data *flow* diagram pada Gambar 11 memiliki dua subsistem lagi yaitu subsistem enkripsi (1.0) dan subsistem dekripsi (2.0).

Aliran data pada subsistem enkripsi adalah pengguna memberi perintah untuk mengakses citra dalam penyimpanan untuk dimasukkan ke dalam subsistem enkripsi kemudian memberikan kunci yang akan digunakan dalam proses enkripsi. Hasil dari subsistem enkripsi adalah citra yang telah terenkripsi.

Subsistem dekripsi merupakan subsistem kedua yang bertujuan untuk mengembalikan citra yang telah terenkripsi menjadi citra semula. Subsistem dekripsi mempunyai sistem kerja yang hampir sama dengan subsistem enkripsi. Pada subsistem dekripsi pengguna dapat mengakses citra pada penyimpanan data langsung dari aplikasi. Pengguna juga memasukkan kunci yang akan diproses dalam dekripsi sehingga akan dihasilkan citra yang sama seperti citra yang belum terenkripsi.



Gambar 11. DFD Level 0

DFD Level 1

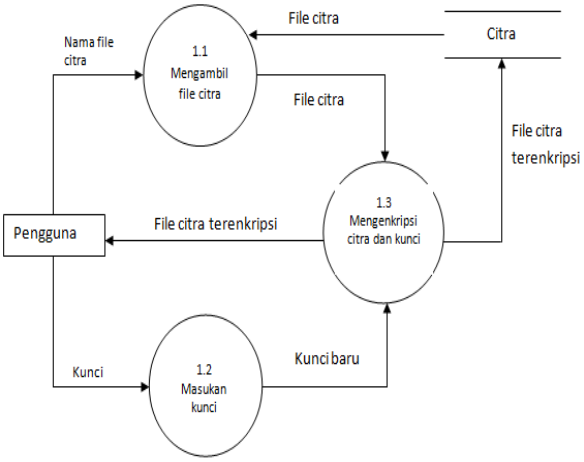
Pada Gambar 12 dan Gambar 13 adalah DFD sistem level 1. DFD system level 1 adalah penjabaran proses data pada DFD level 0. Pada DFD level 0 terdapat dua proses data yaitu subsistem enkripsi dan subsistem dekripsi. Subsistem enkripsi akan dijabarkan pada Gambar 12 sedangkan untuk subsistem dekripsi dijabarkan pada Gambar 13. Untuk penjelasan lebih lanjut

 ISSN: 2338-493X

tentang proses yang terjadi pada Gambar 12 dan Gambar 13, dapat dibaca pada subbab selanjutnya.

Subsistem Enkripsi

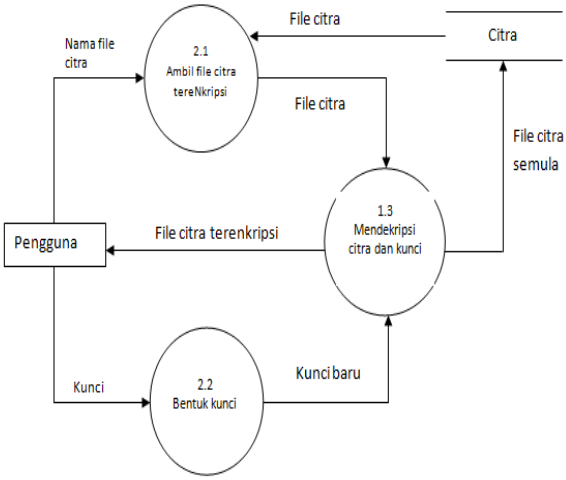
Subsistem enkripsi pada Gambar 12 mempunyai tiga proses yang berada di dalamnya. Proses yang ada di dalam subsistem enkripsi adalah proses pengambilan citra, proses pembentukan kunci serta proses enkripsi terhadap citra dan kunci baru yang telah dibentuk oleh proses pembentukan kunci. Hasil dari enkripsi citra dan kunci akan dapat diakses oleh pengguna langsung dari aplikasi dan pengguna dapat memberi perintah untuk menyimpan citra ke dalam media penyimpanan citra.



Gambar 12. DFD Level 1 Subsistem Enkripsi

Subsistem Dekripsi

Gambar 13 merupakan gambaran sistem yang terdapat pada proses subsistem dekripsi. Proses data pada subsistem dekripsi ada tiga yaitu proses pengambilan citra terenkripsi, proses pembentukan kunci serta proses mendekripsi citra dan kunci. Pengambilan citra pada proses 2.1 pada Gambar 13 merupakan perintah dari pengguna untuk memasukkan citra yang berada pada media penyimpanan untuk dimasukkan ke dalam aplikasi. Proses masukan yang kedua adalah proses pembentukan kunci. Proses membentuk kunci yang dimasukkan ke dalam aplikasi menggunakan algoritma knapsack dan algoritma genetika, sehingga dihasilkan kunci baru yang kemudian menjadi masukan untuk proses 2.3 pada Gambar 13. Proses yang ketiga pada sub sistem enkripsi adalah proses mendekripsi kunci dan citra sehingga dihasilkan citra awal yang sama dengan citra sebelum dienkripsi. Citra yang telah didekripsi akan langsung dapat dilihat oleh pengguna pada aplikasi, dan pengguna dapat memberikan perintah untuk menyimpan citra ke dalam media penyimpanan data.



Gambar 13 DFD Level 1 Subsistem Dekripsi

5. PENGUJIAN DAN ANALISIS

Pengujian yang dilakukan pada penelitian ini adalah pengujian sensitivitas kunci serta pengujian kriptografi terhadap citra dengan ekstensi png, jpeg, dan bmp. Untuk setiap pengujian dapat dilihat pada subbab berikut:

5.1. Pengujian Sensitivitas Kunci

Pengujian sensitivitas kunci merupakan hal yang penting dalam sistem kriptografi. Pada enkripsi citra dengan penggunaan kunci, perubahan kecil pada masukan kunci akan mempengaruhi hasil dekripsi. Pengujian sensitivitas kunci bertujuan untuk melihat perbedaan dari hasil kriptografi berbagai macam kunci. Perbedaan hasil enkripsi yang didapat membuktikan kekuatan *chiperimage* yang tidak dapat didekripsikan dengan kunci yang berbeda. Untuk memperjelas hasil pengujian sensitivitas kunci dapat dilihat pada Tabel 1.

 ISSN: 2338-493X

Tabel 1 Hasil Pengujian Sensitivitas Kunci Pada Citra

Citra Asli	Hasil Enkripsi Dengan Kunci				
	67890657	/%&*7)(!	Algoritma	martinus dias	knapsack dan algen
					
					
					
					
					
					

Citra yang berekstensi png, jpeg, dan bmp. Pengujian dilakukan dengan mengombinasikan 6 buah citra berekstensi png, jpeg, dan bmp dengan 5 buah kunci berbeda. Pada Tabel 1 tampak bahwa kunci mempengaruhi bentuk enkripsi citra, sehingga dapat disimpulkan sensitivitas kunci berlaku pada citra.

Pada Tabel 1, citra terenkripsi berbeda-beda sesuai dengan perubahan kunci yang menjadi masukan. Sehingga hasil enkripsi citra tidak dapat didekripsi dengan menggunakan kunci yang berbeda dari kunci yang digunakan untuk proses enkripsi.

5.2 Pengujian Kriptografi Citra

Pengujian kriptografi citra adalah untuk menguji apakah citra yang telah dienkripsi dapat kembali kebentuk semula pada saat didekripsi. Pengujian dilakukan dengan memasukan sembarang kunci dan citra kemudian dienkripsi. Hasil enkripsi kemudian disimpan untuk dipanggil kembali pada proses dekripsi. Proses dekripsi akan dilakukan dengan cara memasukan kembali citra hasil enkripsi dan kunci yang digunakan pada proses enkripsi.

a. Pengujian Kriptografi Pada Citra Berekstensi Png

Pengujian kriptografi citra berekstensi png bertujuan untuk mengetahui apakah aplikasi kriptografi yang dibangun dapat bekerja pada citra png. Aplikasi dapat dikatakan bekerja dengan baik jika pada proses enkripsi dan dekripsi, akan dihasilkan citra yang samaseperti tampak pada Tabel 2.

Tabel 2. Pengujian Kriptografi citra png

No	Citra Asli	Kunci	Kriptografi		Keterangan
			Enkripsi	Dekripsi	
1		martinus dias			Berhasil (jpeg)
2		Kriptografi			Berhasil (jpeg)
3		Enkripsi			Berhasil (png)
4		Dekripsi			Berhasil (png)
5		knapsack dan algen			Berhasil (bmp)
6		Cob4			Berhasil (bmp)

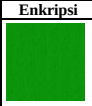
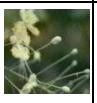
Dari data yang tampak pada Tabel 2, pengujian kriptografi untuk proses enkripsi dan dekripsi berhasil. Citra yang dikombinasikan dengan kunci menghasilkan citra terenkripsi yang disimpan ke dalam ekstensi png. Citra hasil enkripsi kemudian didekripsikan kembali dengan kunci yang sama dengan kunci yang digunakan pada proses enkripsi, sehingga dihasilkan citra yang sama seperti citra semula. Dari hasil pengujian yang tampak pada Tabel 2 dapat disimpulkan bahwa aplikasi kriptografi dapat bekerja dengan baik pada citra berekstensi png.

b. Pengujian Kriptografi Pada Citra Berekstensi Jpeg

Pengujian kriptografi citra berekstensi jpeg bertujuan untuk mengetahui apakah aplikasi dapat berjalan dengan baik pada citra jpeg. Pengujian akan dila-kukan dengan cara menyimpan citra hasil enkripsi ke dalam ekstensi jpeg dan kemudian hasil enkripsi tersebut didekripsikan kembali ke dalam ekstensi jpeg, png, dan bmp. Hasil pengujian dapat dilihat pada Tabel 3.

 ISSN: 2338-493X

Tabel 3 Pengujian Kriptografi Citra jpeg.

No	Citra Asli	Kunci	Kriptografi		Keterangan
			Enkripsi	Dekripsi	
1		martinus dias			Ada *noise* (jpeg)
2		Kriptografi			Ada *noise* (jpeg)
3		Enkripsi			Ada *noise* (png)
4		Dekripsi			Ada *noise* (png)
5		knapsack dan algen			Ada *noise* (bmp)
6		Cob4			Ada *noise* (bmp)

Berdasarkan hasil pengujian pada citra jpeg, pengujian enkripsi berhasil tetapi pada proses dekripsi citra tidak berhasil kembali seperti semula. Citra hasil dekripsi memiliki *noise* sehingga citra tampak tidak sama dengan citra semula.

c. Pengujian Kriptografi Pada Citra Berekstensi bmp

Pengujian kriptografi citra berekstensi bmp bertujuan untuk mengetahui apakah aplikasi dapat berjalan dengan baik pada citra bmp. Pengujian akan dilakukan dengan cara menyimpan citra hasil enkripsi ke dalam ekstensi bmp dan kemudian hasil enkripsi tersebut didekripsikan kembali ke dalam ekstensi bmp. Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4 Pengujian Kriptografi Citra bmp.

No	Citra Asli	Kunci	Kriptografi		Keterangan
			Enkripsi	Dekripsi	
1		martinus dias			Berhasil (jpeg)
2		Kriptografi			Berhasil (jpeg)
3		Enkripsi			Berhasil (png)
4		Dekripsi			Berhasil (png)
5		knapsack dan algen			Berhasil (bmp)
6		Cob4			Berhasil (bmp)

Dari data yang tampak pada Tabel 4, pengujian kriptografi untuk proses enkripsi dan dekripsi berhasil. Citra yang dikombinasikan dengan kunci menghasilkan citra terenkripsi. Citra hasil enkripsi kemudian didekripsikan kembali dengan kunci yang sama dengan kunci yang digunakan pada proses enkripsi sehingga dihasilkan citra yang sama seperti citra semula. Dari hasil pengujian yang tampak pada Tabel 4 dapat disimpulkan bahwa aplikasi kriptografi dapat bekerja dengan baik pada citra berekstensi bmp.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil penelitian yang dilakukan dengan menerapkan metode kriptografi algoritma knapsack, algoritma genetika, dan algoritma catmap pada citra, dapat disimpulkan bahwa:

1. Pengujian kriptografi terhadap semua citra yang semula berekstensi bmp dan png, dapat didekripsikan menjadi citra yang berekstensi bmp, png, dan jpg dengan baik, namun pengujian pada citra yang berekstensi jpeg hasil dekripsinya terdapat *noise*.
2. Hasil enkripsi yang dihasilkan pada kriptografi mempunyai sensitivitas terhadap kunci. Jika kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi, maka tidak dapat mengembalikan citra seperti semula.

6.2 Saran

Saran untuk pengembangan sistem kriptografi menggunakan algoritma knapsack, algoritma genetika, dan algoritma catmap adalah diharapkan peneliti selanjutnya dapat:

1. Menjaga ukuran citra agar tetap sama setelah melalui proses enkripsi dan dekripsi.
2. Dapat mengatasi permasalahan munculnya *noise* pada hasil dekripsi citra berekstensi jpeg.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. 2012. *Algoritma Enkripsi Citra Digital dengan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif Terhadap Bit-Bit MSB.* Bandung: Institut Teknologi

[2]Rahmi, Nitia. 2014. *Penerapaan Konsep Algoritma Genetika Untuk Meningkatkan Aspek Kerahasiaan Data Pada Algoritma Knapsack*. Bandung: Institut Teknologi Bandung.

[3] Kadir, Abdul dan Susanto, Adhi. 2013. Teori dan Aplikasi Pengolahan Citra. Andi: Yogyakarta.

[4] Zukhri, Zainudin. 2014. Algoritma Genetika. Andi: Yogyakarta.

[5] Suyanto. 2005. Algoritma Genetika Dalam Matlab. Andi: Yogyakarta.

[6] Munir, Rinaldi. 2004. *Algoritma Knapsack.* Bandung: Departemen Teknik Informatika Institut Teknologi Bandung.http://www.scribd.com/doc/261797470/Algoritma-Knapsack, di-akses tanggal 19 November 2015

[7] Purba, Ronsen; Halim, Arwin; Syahputra, Indra. *Enkripsi Citra Digital Menggunakan Arnold's Cat Map* dan *Nonlinear Chaotic Algorithm*. Medan: Stimik Mikroskil.

[8] Struss, Katherine. 2009. *A Chaotic Image Encryption.* Morris: University of Minnesota. chinacracy.com/cracker/1425595080_2489007d5e/kate-struss-final.pdf, diakses tanggal 19 November 2015.