IMPLEMENTASI KRIPTOGRAFI *AFFINE CIPHER* PADA CITRA DIGITAL HASIL STEGANOGRAFI METODE *PARITY CODING* DENGAN *PSEUDO RANDOM NUMBER GENERATOR* (PRNG)

SKRIPSI

Oleh

Annash Zaenun Muhendra
NIM 121810101085

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

IMPLEMENTASI KRIPTOGRAFI *AFFINE CIPHER* PADA CITRA DIGITAL HASIL STEGANOGRAFI METODE *PARITY CODING* DENGAN *PSEUDO RANDOM NUMBER GENERATOR* (PRNG)

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Annash Zaenun Muhendra
NIM 121810101085

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. kakek Siswoyo, bapak Ahmad Munif dan ibu Heti Isnaini Agustina, yang telah memberikan cinta dan kasih sayang;
2. adikku tercinta Alfiana Ratmeila;
3. sahabat angkatan 2012 (BATHICS ’12) yang selalu memberikan bantuan dan dukungan yang tulus.
4. keluarga besar mbah masono di seluruh indonesia yang selalu memberikan do’a dan dukungan.
5. Almamater tercinta Jurusan Matematika FMIPA Universitas Jember, SMK MUHAMMADIYAH 2 GENTENG, SMP N 2 CILACAP, SD N 2 CILACAP, TK MAKMUR CILACAP.

MOTTO

”Terkadang seseorang yang di luar dugaan yang mampu melakukan sesuatu di luar dugaan”.[1]

”Kebanggaan kita yang terbesar bukanlah tidak pernah gagal, tetapi bangkit kembali setiap kali kita jatuh”.[2]

[1]Profesor Alan Turing
[2]Confusius

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

nama : Annash Zaenun Muhendra

NIM : 121810101085

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ”Implementasi Kriptografi *Affine Cipher* pada Citra Digital Hasil Steganografi Metode *Parity Coding* dengan *Pseudo Random Number Generator* (PRNG)” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juni 2016

Yang menyatakan,

Annash Zaenun Muhendra

NIM 121810101085

SKRIPSI

IMPLEMENTASI KRIPTOGRAFI *AFFINE CIPHER* PADA CITRA DIGITAL HASIL STEGANOGRAFI METODE *PARITY CODING* DENGAN *PSEUDO RANDOM NUMBER GENERATOR* (PRNG)

Oleh
Annash Zaenun Muhendra
NIM 121810101085

Pembimbing:

Dosen Pembimbing 1 : Ahmad Kamsyakawuni, S.Si., M.Kom.
Dosen Pembimbing 2 : M. Ziaul Arif, S.Si., M.Sc.

PENGESAHAN

Skripsi berjudul ”Implementasi Kriptografi *Affine Cipher* pada Citra Digital Hasil Steganografi Metode *Parity Coding* dengan *Pseudo Random Number Generator* (PRNG)” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,	Sekretaris,
Ahmad Kamsyakawuni, S.Si., M.Kom.	M. Ziaul Arif, S.Si., M.Sc.
NIP. 197211291998021001	NIP. 198501112008121002
Penguji I,	Penguji II,
Drs. Rusli Hidayat, M.Sc.	Kusbudiono, S.Si., M.Si.
NIP. 196610121993031001	NIP. 197704302005011001

Mengesahkan

Dekan

Drs. Sujito, Ph.D.

NIP. 196102041987111001

RINGKASAN

Implementasi Kriptografi *Affine Cipher* pada Citra Digital Hasil Steganografi Metode *Parity Coding* dengan *Pseudo Random Number Generator* (PRNG); Annash Zaenun Muhendra, 121810101085; 2016; 54 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan kemajuan teknologi informasi saat ini semakin memudahkan para pelaku kejahatan komputer dengan menyalahgunakan teknologi komputer untuk mendukung kegiatannya, dimana kegiatan tersebut sangat mengganggu privasi seseorang.

Oleh karena itu diperlukan sebuah sistem atau aplikasi sebagai pengaman data. Salah satu sistem atau aplikasi yang diperlukan sebagai pengaman suatu data adalah kriptografi. Kriptografi adalah ilmu mengenai teknik mengacak suatu data menjadi sesuatu yang sulit dibaca. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih kearah metode-metode yang digunakan. Salah satu metode yang akan dibahas dalam penelitian ini adalah kriptografi dengan algoritma *Affine Cipher*. Namun, menurut Kromodimoeljo (2009), kriptografi dengan algoritma *Affine Cipher* masih tergolong mudah untuk dilakukan analisa frekuensi terkait pencarian kunci enkripsi. Untuk menghindari hal tersebut, maka pesan yang telah dienkripsi disembunyikan ke dalam suatu obyek. Steganografi adalah ilmu untuk menyembunyikan informasi yang merupakan cara untuk mencegah pendeteksian pesan tersembunyi.

Data yang digunakan dalam penelitian ini adalah data teks berupa pesan rahasia. Karakter-karakter dari pesan tersebut merupakan karakter yang terdapat pada *keyboard* (ASCII *printable character*). Selain data teks, data yang digunakan adalah gambar (*image*) berekstensi .bmp, .png, .gif. Tugas akhir ini, dilakukan proses pengamanan pesan yang berupa teks menggunakan gabungan kriptografi dan steganografi. *Plaintext* yang berupa teks dienkripsi menggunakan algoritma *Affine Cipher* sehingga dihasilkan *ciphertext*. *Ciphertext* tersebut kemudian diubah ke dalam desimal sesuai kode ASCII

yang selanjutnya diubah ke bilangan biner. Biner yang dibentuk dari *ciphertext* kemudian disisipkan ke citra digital menggunakan steganografi metode *parity coding*, namun penyisipan yang dilakukan berbeda dengan metode *party coding* pada umumnya, yaitu dengan menentukan posisi piksel atau region untuk menyisipkan pesan menggunakan *Pseudo Random Number Generator* (PRNG). Setelah disisipkan, citra hasil penyisipan dienkripsi menggunakan kriptografi *Affine Cipher* dengan pasangan kunci yang sama dengan proses enkripsi *plaintext* dan dihasilkan citra kripto. Citra kripto inilah yang kemudian dikirim ke pihak kedua untuk dibaca isi pesan tersebut. Proses pembacaan pesan yang pertama adalah dengan proses dekripsi citra menggunakan kriptografi *Affine Cipher* dan dihasilkan *stego object*. Pesan yang telah disisipkan pada *stego object* diekstrak kembali dan dihaslikan *hiddentext* yang berisi *ciphertext*, kunci a, b dan $seeds$. Hasil dari *hiddentext* tersebut didekripsi sehingga dihasilkan *plaintext*.

Waktu komputasi yang dibutuhkan untuk menyisipkan pesan bergantung pada panjang pesan yang disisipkan. Semakin besar panjang pesan, maka semakin banyak waktu yang dibutuhkan untuk menyisipkan pesan. Hal ini berlaku juga untuk proses ekstraksi pesan. Hasil enkripsi gambar dipengaruhi oleh pemilihan kunci a, semakin besar nilai kunci a, maka semakin baik citra kripto yang dihasilkan (tidak terdeteksi gambar awalnya). Ukuran file dari citra kripto yang dihasilkan sama dengan ukuran *cover object* atau tidak mengalami perubahan ukuran file. Waktu komputasi yang dihasilkan untuk proses enkripsi dan dekripsi tidak berbedah jauh yaitu 0.2891 untuk proses enkripsi dan 0.298694 untuk proses dekripsi serta ukuran file tidak pengaruh terhadap proses enkripsi dan dekripsi gambar.

PRAKATA

Puji syukur ke hadirat Allah Swt atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul ”Implementasi Kriptografi *Affine Cipher* pada Citra Digital Hasil Steganografi Metode *Parity Coding* dengan *Pseudo Random Number Generator* (PRNG)”. Tugas akhir ini disusun untuk memenuhi salah satu syarat pada program pendidikan strata satu (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Pada kesempatan ini penulis mengucapkan terima kasih atas bantuan dan bimbingan dalam penyusunan tugas akhir ini, terutama kepada yang terhormat:

1. Drs. Sujito, Ph.D., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
2. Kusbudiono, S.Si., M.Si., selaku Ketua Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
3. Ahmad Kamsyakawuni, S.Si., M.Kom, selaku Dosen Pembimbing Utama dan M. Ziaul Arif, S.Si., M.Sc. selaku Dosen Pembimbing Anggota;
4. Kedua orang tua tercinta, ibu Heti Isnaeni Agustina dan ayah Ahmad Munif serta kakek, adek dan keluarga besar yang selalu memberikan dukungan dan doa;
5. Dosen dan Karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
6. Teman-teman satu angkatan yang telah memberikan banyak kenangan dan dukungan.

Semoga bantuan, bimbingan, dan dorongan beliau dicatat sebagai amal baik oleh Allah SWT dan mendapat balasan yang sesuai dari-Nya. Selain itu, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan penyusunan tugas akhir ini. Akhirnya penulis berharap, semoga tugas akhir ini dapat bermanfaat.

Jember, Juni 2016 Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan kemajuan teknologi informasi saat ini semakin memudahkan para pelaku kejahatan komputer dengan menyalahgunakan teknologi komputer untuk mendukung kegiatannya, dimana kegiatan tersebut sangat mengganggu privasi seseorang. Pesan yang seharusnya hanya diketahui oleh pihak yang berhak mengetahui pesan tersebut, namun dapat diketahui oleh seseorang yang tidak atau bahkan dilarang mengetahuinya. Apabila hal tersebut terjadi, tentu saja sangat mengganggu atau bahkan sangat berbahaya jika pesan tersebut berisi sesuatu yang sangat penting yang menyangkut kelangsungan hidup seseorang atau suatu instansi. Oleh karena itu diperlukan sebuah sistem atau aplikasi sebagai pengaman data.

Salah satu sistem atau aplikasi yang diperlukan sebagai pengaman suatu data adalah kriptografi. Kriptografi adalah ilmu mengenai teknik mengacak suatu data menjadi sesuatu yang sulit dibaca. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih kearah metode-metode yang digunakan. Salah satu metode yang akan dibahas dalam penelitian ini adalah kriptografi dengan algoritma *Affine Cipher*. Namun, menurut Kromodimoeljo (2009), kriptografi dengan algoritma *Affine Cipher* masih tergolong mudah untuk dilakukan analisa frekuensi terkait pencarian kunci enkripsi. Untuk menghindari hal tersebut, maka pesan yang telah dienkripsi disembunyikan ke dalam suatu obyek. Steganografi adalah ilmu untuk menyembunyikan informasi yang merupakan cara untuk mencegah pendeteksian pesan tersembunyi.

Purba (2012) membahas Implementasi Penyembunyuian dan Penyandian Pesan pada Citra Menggunakan Algoritma *Affine Cipher* dan Metode *Least Significant Bit*. Pada proses penelitiannya, data berupa teks diinputkan secara langsung oleh pengguna pada program aplikasi kemudian diubah ke dalam naskah acak yang selanjutnya disisipkan ke dalam citra digital yang disebut dengan *Stego Object*. Pesan yang terdapat

pada *Stego Object* diekstrak kembali sehingga didapatkan naskah acak kemudian di ubah ke dalam naskah asli.

Penelitian yang dilakukan Purba (2012), data disisipkan pada *header* citra secara berurutan serta tidak dilakukan proses enkripsi citra sehingga tingkat kesulitan untuk memecahkan pesan masih rendah. Oleh karena itu, pada tugas akhir ini, penyisipan karakter dari *ciphertext* dilakukan secara acak menggunakan *Pseudo Random Number Generator* (PRNG) metode *Linear Congruential Generator* (LCG). Karena dengan algoritma LCG dapat menghasilkan deretan bilangan acak dengan waktu yang cepat. Hal ini disebabkan LCG hanya membutuhkan sedikit operasi bit (Susanti, 2007). Sedangkan penyisipan pesan menggunakan metode *Parity Coding*. Metode ini merupakan modifikasi dari metode LSB yang sebelumnya digunakan untuk menyisipkan pesan pada audio oleh Susanti (2007). Kemudian setelah pesan disisipkan secara acak, citra digital hasil penyisipan dienkripsi dengan algoritma *Affine Cipher*. Dengan melakukan proses penyisipan secara acak dan metode penyisipan yang dimodifikasi serta citra hasil penyisipan dienkripsi, diharapkan akan meningkatkan kemanan pesan yang disembunyikan.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah :

a. Bagaimana penggunaan PRNG dalam penyisipan pesan ?
b. Bagaimana menganalisis citra hasil penyisipan ?
c. Bagaimana hasil enkripsi citra untuk menyembunyikan gambar dan pesan ?
d. Bagaimana mengembalikan gambar dan pesan ?

1.3 Batasan Masalah

Adapun batasan masalah dari penulisan tugas akhir ini adalah :

a. File yang akan disisipkan berbentuk teks.

b. Media digital penampungnya berupa file gambar (image) dengan ekstensi .bmp, .png, dan .gif.

c. Karakter yang akan disisipkan adalah ASCII *printable character*.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah :

a. Menjelaskan penggunaan PRNG dalam penyisipan pesan.

b. Menganalisis kualitas dan keamanan citra digital hasil penyisipan.

c. Menyembunyikan gambar dan pesan.

d. Menganalisis kualitas gambar dan pesan hasil pengembalian kembali gambar dan pesan.

e. Membuat program aplikasi keamanan data sebagai media komunikasi yang bersifat rahasia.

1.5 Manfaat

Manfaat yang diharapkan dari penulisan tugas akhir ini antara lain:

a. Mengurangi kemungkinan pesan yang disembunyikan terdeteksi karena letak penyisipan pesan pada posisi acak.

b. Mengetahui kemampuan PRNG dalam penyembunyian pesan.

c. Mengetahui gambar hasil enkripsi dengan algoritma *Affine Cipher*.

BAB 2. TINJAUAN PUSTAKA

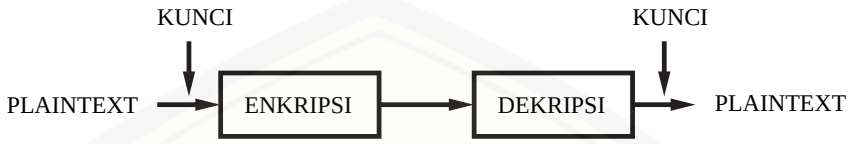
2.1 Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: *cryptos* artinya *secret* (rahasia), sedangkan *graphein* artinya *writing* (tulisan), Jadi, kriptografi berarti *secret writing* (tulisan rahasia). Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Definisi yang digunakan di dalam buku menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, *diplomat*, dan mata-mata. Namun saat ini kriptografi lebih dari sekadar *privacy*, tetapi juga untuk tujuan data *integrity*, *authentication*, dan *non-repudation* (Munir, 2006).

Terdapat 2 bagian penting dalam proses kriptografi yaitu enkripsi dan dekripsi. Enkripsi merupakan hal yang sangat penting supaya keamanan data yang dikirimkan bisa terjaga kerahasiaannya. Enkripsi bisa diartikan dengan chiper atau kode, di mana pesan asli (*plaintext*) diubah menjadi kode-kode tersendiri sesuai metode yang disepakati oleh kedua belah pihak, baik pihak pengirim pesan maupun penerima pesan (Pramono, 2009).

Informasi asal yang dapat dimengerti disimbolkan oleh *plaintext*, kemudian oleh algoritma enkripsi diterjemahkan menjadi informasi acak yang tidak dimengerti dan disimbolkan dengan *chipertext*. Proses enkripsi terdiri dari algoritma dan kunci. Kunci merupakan suatu *string bit* pendek yang mengontrol algoritma. Algoritma enkripsi akan memberikan hasil yang berbeda tergantung pada kunci yang digunakan. Mengubah kunci dari enkripsi akan mengubah *output* dari algoritma enkripsi. Setelah itu *ciphertext* kemudian ditransmisikan oleh pengirim. Setelah itu dilakukan proses dekripsi,

yaitu proses untuk mengembalikan teks yang telah acak ke bentuk semula dengan algoritma dan kunci yang sama, dalam hal ini dilakukan oleh penerima sehingga akan kembali menjadi sebuah informasi yang dapat dipahami oleh penerima (Ariyus, 2006).



Gambar 2.1 Proses Enkripsi dan Dekripsi
(sumber: Pramono, 2009)

2.1.1 Jenis - Jenis Serangan

Serangan (*attack*) adalah setiap usaha (*attempt*) atau percobaan yang dilakukan oleh kriptanalis untuk menemukan kunci dan mengungkap *plaintext* (Munir, 2006). Beberapa jenis serangan yang dapat dilakukan oleh kriptanalis, dengan asumsi bahwa kriptanalis telah mengetahui algoritma kriptografi yang digunakan, antara lain :

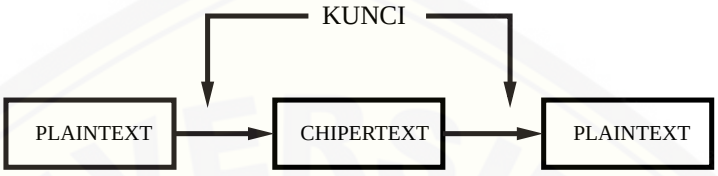
a. *Ciphertext Only Attack*, yaitu penyerang hanya mendapatkan pesan yang telah dirahasiakan.
b. *Known Plaintext Attack*,yaitu dimana selain mendapatkan sandi, penyerang juga mendapatkan pesan asli.
c. *Chosen Plaintext Attack*, sama dengan known plaintext attack, namun penyerang bahkan dapat memilih penggalan dari pesan asli yang akan disandikan.

2.1.2 Algoritma Kriptografi

Algoritma kriptografi disebut juga cipher yaitu aturan untuk *enchipering* dan *dechipering*, atau fungsi yang digunakan untuk enkripsi dan dekripsi. Beberapa cipher memerlukan algoritma yang berbeda untuk *enciphering* dan *dechipering*. Dalam kriptografi terdapat dua macam algoritma kriptografi, yaitu

a. Algoritma Simetris

Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional. Algoritma ini menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi. Pada skema enkripsi *symmetric-key*, digunakan sebuah kunci untuk melakukan proses enkripsi dan dekripsinya.



Gambar 2.2 Prosedur Kerja Algoritma Simetris

b. Algoritma Asimetris

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga Algoritma Kunci Umum (*Public Key Algorithm*) karena kunci untuk enkripsi dibuat umum (*publik key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut Kunci Pribadi (*Private Key*).

Gambar 2.3 Prosedur Kerja Algoritma Asimetris *Private-key*

2.2 Aritmatika Modular

Aritmatika modular sangat berperan dalam kriptografi karena banyak digunakan dalam algoritma enkripsi, baik untuk enkripsi simetris maupun untuk *public-key cryptography*. Dalam aritmatika modular, konsep gcd digunakan antara lain untuk operasi *inverse*. Gcd dapat dikalkulasi secara efisien menggunakan algoritma *Euclid*, algoritma sangat penting yang telah berusia lebih dari 2000 tahun.

Untuk setiap pasangan bilangan bulat a dan b, jika terdapat bilangan bulat q sehingga $a = qb$, maka b membagi a, dan b disebut pembagi (*divisor* atau faktor) dari a dengan notasi $b|a$. Notasi $b \nmid a$ digunakan jika b bukan pembagi a.

Definisi 2.2.1. *Jika $d|a$ dan $d|b$ maka d adalah pembagi persekutuan (common divisor) dari a dan b. Untuk setiap pasangan bilangan bulat a dan b kecuali jika $a = b = 0$, pembagi persekutuan terbesar (greatest common divisor atau gcd) dari a dan b adalah bilangan bulat unik d dimana:*

1. *d merupakan pembagi persekutuan dari a dan b,*
2. *jika c merupakan pembagi persekutuan dari a dan b, maka $c \leq d$.*

Satu cara untuk mendapatkan $gcd(a, b)$ adalah dengan membuat daftar semua faktor dari a, membuat daftar semua faktor dari b, dan kemudian mencari faktor terbesar yang ada dalam kedua daftar. Akan tetapi, untuk bilangan yang sangat besar, membuat daftar faktor bukanlah sesuatu yang mudah. Ada cara yang jauh lebih efisien untuk mendapatkan $gcd(a, b)$ yaitu menggunakan algoritma *Euclid* (*Euclidean algorithm*). Berikut merupakan teorema mengenai algoritma *Euclidean*.

Teorema 2.2.1 (**Algoritma *Euclidean***).

$$\text{jika } a = qb + r, \text{ maka } gcd(a, b) = gcd(b, r)$$

Algoritma *Euclid* menggunakan rumus diatas secara berulang untuk mendapatkan gcd, yaitu dengan memperkecil kedua bilangan yang dijadikan patokan untuk gcd setiap kali mengulang, tanpa merubah nilai gcd itu sendiri.

Diberikan dua buah bilangan bulat tak-negatif a dan b ($b \leq a$), maka algoritma *Euclidean* untuk mencari pembagi bersama terbesar dari a dan b adalah

Algoritma 1 Euclidean

1: **Input** : a, b
2: **Output** : a
3: **while** $b \neq 0$ **do**
4: $\quad r = a \bmod b$;
5: $\quad a = b$;
6: $\quad b = r$;
7: **end while**

Definisi 2.2.2 (Modulo).

$$a \bmod n = r = a - nq$$

dengan kata lain a mod n adalah *remainder* atau *residue* dari pembagian a oleh n.

Jadi operasi perkalian modulo n dapat dipandang sebagai operasi aritmatika bilangan bulat yang dilanjutkan dengan operasi mod pada hasil operasi bilangan bulat. Rumus untuk perkalian $x.y$ menjadi:

$$x.y = xy \bmod n \tag{2.1}$$

Definisi untuk inverse perkalian b^{-1} adalah:

$$bb^{-1} = 1$$

Jadi b^{-1} adalah bilangan bulat yang memenuhi persamaan:

$$(bb^{-1}) \bmod n = 1 \text{ dengan } 0 \leq b^{-1} < n$$

Teorema 2.2.2 (Inverse). *Suatu bilangan a mempunyai inverse modulo n jika dan hanya jika gcd(a, n) = 1.*

Berdasarkan persamaan (2.1), maka dapat digunakan untuk mendapatkan *inverse* dengan rumus yaitu :
Misalkan x bilangan bulat maka :

$$x^{-1} = (1 + kn)/x \tag{2.2}$$

untuk k dari $1, 2$ sampai x^{-1} berupa bilangan bulat (Purba, 2012).

2.3 Algoritma *Affine Cipher*

Affine cipher adalah teknik cipher yang merupakan perluasan dari *Caesar cipher*. *Affine cipher* tergolong dalam algoritma klasik yang merupakan algoritma penyandian yang sudah ada sebelum era digital sekarang ini. Algoritma klasik pada dasarnya hanya terdiri dari cipher subtitusi dan cipher tranposisi. Cipher subtitusi yaitu proses mensubtitusi karakter-karakter yang ada pada plaintext. Sedangkan cipher tranposisi yaitu proses pertukaran huruf-huruf yang terdapat dalam suatu string.

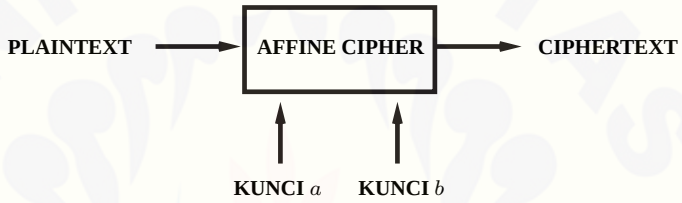
Affine cipher merupakan metode kriptografi yang menggunakan kunci simetris, yang mana kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk dekripsi.

2.3.1 Teknik Enkripsi pada *Affine Cipher*

Proses enkripsi menggunakan *Affine cipher* membutuhkan 2 buah kunci yaitu kunci 1(a) dan kunci 2(b) untuk dapat menghasilkan *ciphertext*. *Plaintext* (P_i) akan dikonversikan menggunakan table konversi sehingga menjadi bentuk decimal, kemudian *ciphertext* (C_i) akan diperoleh dengan mengenkripsi *plaintext* dengan persamaan:

$$C_i = aP_i + b(\text{mod } 26) \tag{2.3}$$

C_i merupakan *ciphertext* dari pergeseran karakter yang terdapat pada plaintext. P_i merupakan pergeseran karakter pada *plaintext*. a merupakan kunci berupa bilangan bulat yang relatif prima dengan 26, apabila a tidak relatif prima dengan 26 maka dekripsi tidak akan bisa dilakukan karena tidak bisa mendapatkan *inverse* dari kunci a berdasarkan teorema (2.3.2). Sedangkan kunci b merupakan pergeseran nilai relatif prima dari a. Agar dapat memperoleh *ciphertext* maka perlu dilakukan perhitungan dengan persamaan (2.3) adapun hasil yang diperoleh masih berupa bilangan decimal, kemudian dari bilangan decimal tersebut akan dikonversi dikonversi menggunakan tabel menjadi *ciphertext*



Gambar 2.4 Proses Enkripsi *Affine Cipher*

Gambar 2.4 menjelaskan bahwa untuk memperoleh *ciphertext* menggunakan *Affine cipher* dibutuhkan input berupa *plaintext* yang akan dienkripsi menggunakan dua kunci.

Contoh 2.3.1. *Plaintext* = BACA; $a = 7$ dan $b = 2$;. Ubahlah Plainteks ke Cipherteks menggunakan algoritma *Affine Cipher*

Penyelesaian :

Misal $n = 26$, maka sebelumnya dicek $gcd(n, a) = 1$ (relatif prima) menggunakan algoritma 1 (*Euclidean*), yaitu:

Tahap 1: $26 \bmod 7 = 5$

Tahap 2: $7 \bmod 5 = 2$

Tahap 3: $5 \bmod 2 = 1$

Tahap 4: $2 \bmod 1 = 0$

Tahap 5: $n = 1$, $a = 0$.

Karena nilai $a = 0$, maka proses berhenti dan $n = 1$ adalah $gcd(n, a)$, sehingga $gcd(n, a) = 1$ (relatif prima).

Kemudian ubah tiap karakter *plaintext* ke dalam nilai decimal dimana nilai tiap karakter ditentukan oleh tabel konversi decimal berikut :

Tabel 2.1 Tabel Konversi Desimal

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

dari tabel didapatkan :

B	A	C	A
1	0	2	0

kemudian dihitung nilai *ciphertext* berdasarkan persamaan (2.3) :

$C[B]$ = (7*1+2) mod 26 = 9

$C[A]$ = (7*0+2) mod 26 = 2

$C[C]$ = (7*2+2) mod 26 = 16

$C[A]$ = (7*0+2) mod 26 = 2

Setelah didapat nilai *ciphertext*, kemudian diubah menjadi karakter dengan mensubsitusi nilai berdasarkan tabel ke jenis karakter. Sehingga *ciphertext* yang dihasilkan adalah J C Q C.

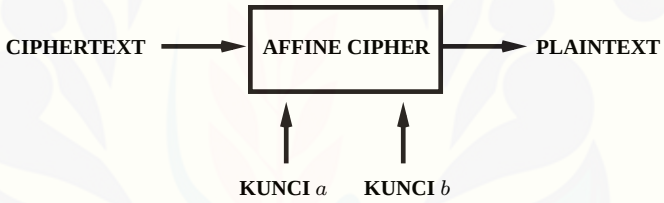
2.3.2 Teknik Dekripsi pada *Affine Cipher*

Proses dekripsi menggunakan *Affine cipher* membutuhkan dua buah kunci yang mana kedua kunci yang dipakai haruslah sama dengan kunci yang digunnakan pada proses enkripsi. Agar dapat memperoleh plaintext maka kunci 1(a) akan dirubah dalam bentuk invers a (mod 26), dinyatakan dengan a^{-1}. Jika a^{-1} ada, maka dekripsi akan

dilakukan dengan persamaan

$$P_i = a^{-1}(C_i - b) \bmod 26 \tag{2.4}$$

P_i merupakan *plaintext* dari pergeseran karakter yang terdapat pada *ciphertext*. C_i merupakan pergeseran karakter karakter pada *ciphertext*. Sedangkan a dan b merupakan kunci yang sama dengan kunci yang digunakan pada proses enkripsi. Agar dapat memperoleh *plaintext* maka diperlukan perhitungan menggunakan persamaan (2.4). Sebelum melakukan perhitungan terlebih dahulu P_i dan C_i harus dikonversikan kedalam bentuk decimal berdasarkan tabel 2.1. Hasil dari perhitungan yang dilakukan akan berbentuk bilangan decimal yang kemudian akan dikonversi menggunakan tabel 2.1 untuk memperoleh *plaintext*.



Gambar 2.5 Proses Dekripsi *Affine Cipher*

Gambar 2.5 menjelaskan bahwa untuk memperoleh *plaintext* menggunakan *Affine cipher* dibutuhkan input berupa *ciphertext* yang akan dienkripsi menggunakan dua buah kunci.

Kekuatan dari *Affine Cipher* ini terletak pada kunci yang dipakai. Kunci ini merupakan nilai integer yang menunjukkan pergeseran karakter-karakter. Selain itu *Affine Cipher* juga menggunakan barisan bilangan-bilangan yang berfungsi sebagai pengali kunci. Barisan yang digunakan dapat berupa bilangan tertentu seperti deret bilangan genap, deret bilangan ganjil, deret bilangan prima, deret fibonaci dapat juga deret bilangan yang dibuat sendiri. Dengan adanya kemungkinan pemilihan kunci yang dipilih lebih bervariatif dan lebih banyak algoritma enkripsi subtitusi lain menjadikan *Affine*

Cipher sebagai sistem enkripsi yang paling sempurna dibandingkan dengan algoritma enkripsi subtitusi lainnya.

Contoh 2.3.2. *Ciphertext* = J C Q C; $a = 7$ dan $b = 2$;. Ubahlah *Chipertext* ke *Plaintext* menggunakan algoritma *Affine Cipher*

Penyelesaian :

Pertama dihitung nilai a^{-1} berdasarkan persamaan (2.2) , maka :

$$a^{-1} = (1 + kn)/a$$

dengan nilai k dimulai dari 0 dan nilainya terus bertambah hingga didapat nilai m^{-1} berupa bilangan bulat positif pertama. Berikut merupakan langkah nendapatkan nilai invers m :

Tahap 1, k = 0: $a^{-1} = (1 + 0 * 26)/7 = 3.8571$

Tahap 2, k = 1: $a^{-1} = (1 + 1 * 26)/7 = 7.5714$

Tahap 3, k = 2: $a^{-1} = (1 + 2 * 26)/7 = 11.2857$

Tahap 4, k = 3: $a^{-1} = (1 + 3 * 26)/7 = 15$

Sehingga didapatkan $a^{-1} = 15$. Selanjutnya ubah *ciphertext* ke decimal sesuai dengan tabel 2.1, sehingga didapatkan :

J	C	Q	C
9	2	16	2

kemudian dilakukan perhitungan berdasarkan persamaan (2.4) untuk mendapatkan nilai decimal dari *plaintext* :

$$P[J] = 15(9-2) \bmod 26 = 1$$
$$P[C] = 15(2-2) \bmod 26 = 0$$
$$P[Q] = 15(16-2) \bmod 26 = 2$$
$$P[C] = 15(2-2) \bmod 26 = 0$$

Setelah dikonversi berdasarkan tabel 2.1 dihasilkan *plaintext*: BACA.

2.4 Citra RGB (Citra Warna atau *truecolor*)

Citra berwarna yaitu citra yang nilai *pixel*-nya merepresentasikan warna tertentu. Banyaknya warna yang mungkin digunakan bergantung kepada kedalaman *pixel* citra yang bersangkutan. Citra RGB direpresentasikan dalam beberapa kanal yang menyatakan komponen-komponen warna penyusun. Banyak kanal yang digunakan bergantung pada model warna yang digunakan pada citra tersebut.

Gambar 2.6 Citra RGB dengan ukuran $h \times w$ *pixel*

Intensitas suatu titik pada citra berwarna merupakan kombinasi dari tiga intensitas :

1. derajat keabuan merah ($f_r(x, y)$)
2. derajat keabuan hijau ($f_g(x, y)$)
3. derajat keabuan biru ($f_b(x, y)$)

2.5 Kode ASCII (*American Standart Code For Information Interchange*)

Kode ASCII merupakan standar internasional dalam kode huruf dan simbol yang bersifat universal. Kode ASCII digunakan oleh komputer dan alat komunikasi lainnya untuk menunjukkan teks (Rachmawanto, 2010).

Karakter kode 32 sampai 125 yang berisi huruf, digit, tanda baca dan beberapa simbol lain yang sebagian besar ditemukan di *keyboard* disebut ASCII *printable character*. ASCII *printable character* dapat dilihat pada tabel 2.2.

Tabel 2.2 Tabel ASCII *printable character*

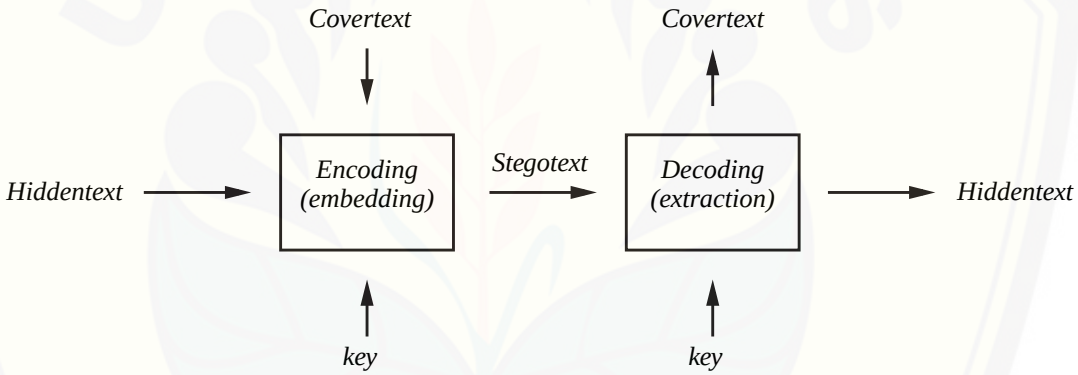
ASCII *code*	*Binnary Code*	*character*	ASCII *code*	*Binnary Code*	*character*
032	00100000	spasi	080	01010000	P
033	00100001	!	081	01010001	Q
034	00100010	”	082	01010010	R
035	00100011	#	083	01010011	S
036	00100100	$	084	01010100	T
037	00100101	%	085	01010101	U
038	00100110	&	086	01010110	V
039	00100111	’	087	01010111	W
040	00101000	(	088	01011000	X
041	00101001	)	089	01011001	Y
042	00101010	*	090	01011010	Z
043	00101011	+	091	01011011	[
044	00101100	,	092	01011100	\
045	00101101	-	093	01011101	]
046	00101110	.	094	01011110	^
047	00101111	/	095	01011111	_
048	00110000	0	096	01100000	~
049	00110001	1	097	01100001	a
050	00110010	2	098	01100010	b
051	00110011	3	099	01100011	c
052	00110100	4	100	01100100	d
053	00110101	5	101	01100101	e
054	00110110	6	102	01100110	f
055	00110111	7	103	01100111	g

ASCII *code*	*Binnary Code*	*character*	ASCII *code*	*Binnary Code*	*character*
056	00111000	8	104	01101000	h
057	00111001	9	105	01101001	i
058	00111010	:	106	01101010	j
059	00111011	;	107	01101011	k
060	00111100	<	108	01101100	l
061	00111101	=	109	01101101	m
062	00111110	>	110	01101110	n
063	00111111	?	111	01101111	o
064	01000000	@	112	01110000	p
065	01000001	A	113	01110001	q
066	01000010	B	114	01110010	r
067	01000011	C	115	01110011	s
068	01000100	D	116	01110100	t
069	01000101	E	117	01110101	u
070	01000110	F	118	01110110	v
071	01000111	G	119	01110111	w
072	01001000	H	120	01111000	x
073	01001001	I	121	01111001	y
074	01001010	J	122	01111010	z
075	01001011	K	123	01111011	{
076	01001100	L	124	01111100	|
077	01001101	M	125	01111101	}
078	01001110	N	126	01111101	~
079	01001111	O	127	01111101	Δ

2.6 Steganografi

Steganografi berasal dari bahasa Yunani yaitu Steganos yang berarti menyembunyikan dan Graptos yang artinya tulisan sehingga secara keseluruhan artinya adalah tulisan yang disembunyikan. Secara umum steganografi merupakan seni atau ilmu yang digunakan untuk menyembunyikan pesan rahasia dengan segala cara sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari pesan rahasia tersebut (Munir, 2006).

Penyisipan pesan ke dalam media *covertext* dinamakan *encoding*, sedangkan ekstraksi pesan dari *stegotext* dinamakan *decoding*. Kedua proses ini memerlukan kunci rahasia agar hanya pihak yang berhak saja yang dapat melakukan penyisipan dan ekstraksi pesan. Berikut merupakan ilustrasi proses *Encoding* dan *Decoding* pada steganografi.



Gambar 2.7 Proses *Encoding* dan *Decoding* Pesan

Di dalam steganografi citra digital, *hiddentext* atau *embedded message* merupakan teks yang akan disisipkan ke dalam *covertext* atau *cover object*, yaitu file yang digunakan sebagai media penampung pesan yang akan disisipkan. Hasil dari *encoding* atau *embedding* pesan ke dalam file citra akan dihasilkan *stegotext* atau *stego object* yang merupakan file yang berisikan pesan yang telah disisipkan.

2.7 Steganografi *Parity Coding*

Sistem kerja steganografi dengan metode *parity coding* adalah membagi sinyal media yang telah di encode menjadi beberapa region terpisah dengan ukuran statis. *Parity* bit dari setiap region dapat disesuaikan sesuai dengan panjang bit pesan rahasia yang sudah di enkripsi. Bit dari pesan rahasia akan disisipkan secara merata ke dalam region yang ada. Jika bit yang akan dimasukkan ke dalam region nilainya berbeda, maka susunan dari bit-bit LSB harus diubah sedemikian rupa sehingga *parity bit region* nilainya sama dengan bit pesan rahasia yang akan disisipkan, namun jika nilainya sama, region tidak perlu diubah.

Proses pertama yang dilakukan dalam penyembunyian pesan adalah membagi media stegano yang dalam hal ini adala citra digital kedalam region-region. Banyaknya region ini ditentukan oleh panjang isi berkas yang dijadikan objek stegano.

Misalkan nilai biner dari karakter yang akan disembunyikan adalah 10110010 dan hasil encode media citra digital adalah sebagai berikut.

10100011	00010010	11001000
00100111	11001000	11101001
11001000	00100111	11101001

kemudian diproses secara parity dalam region masing-masing.

bit 1 = 1

region 1 = 10100011

parity bit region 1→ $1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 0$

karena nilai bit pertama pesan tidak sama dengan *parity* bit region 1, maka bit-bit pada region 1 dimanipulasi agar nilai *parity* bit sama dengan bit pesan. Manipulasi yang

dilakukan adalah dengan menukar LSB dari region. Sehingga region 1 yang baru yaitu.

region 1 baru : 1010001-**0**

parity bit region 1 baru $\rightarrow$ 1 $\oplus$ 0 $\oplus$ 1 $\oplus$ 0 $\oplus$ 0 $\oplus$ 0 $\oplus$ 1 $\oplus$ **0** = 1

bit 2 = 0

region 2 = 00010010

parity bit region 2$\rightarrow$ 0 $\oplus$ 0 $\oplus$ 0 $\oplus$ 1 $\oplus$ 0 $\oplus$ 0 $\oplus$ 1 $\oplus$ 0 = 0

karena nilai bit kedua dari pesan sama dengan *parity* bit region 2, maka region tidak perlu dimodifikasi. Proses ini dilakukan keseluruh bit-bit pada pesan dalam region-region yang telah disediakan. Sedangkan untuk proses ekstraksi pesan, dilakukan proses kebalikan dari proses penyembunyian pesan.

2.8 *Pseudo Random Number Generator* (PRNG)

Pseudo Random Number Generator adalah algoritma yang membangkitkan deretan bilangan yang tidak benar-benar acak. Bilangan acak dihasilkan dengan rumus-rumus matematika dan dapat berulang kembali secara periodik. Salah satu algoritma PRNG yang sering digunakan adalah *Linear Congruential Generator* (LCG). *Linear Congruential Generator* (LCG) adalah salah satu pembangkit bilangan acak tertua dan sangat terkenal. LCG didefinisikan dalam bentuk :

$$X_n = (a1X_{n-1} + b1) \bmod m \tag{2.5}$$

Keterangan :

X_n = bilangan acak ke-n dari deretnya

X_{n-1} = bilangan acak sebelumnya

$a1$ = faktor pengali

$b1$ = increment

m = modulus

($a1$, $b1$, dan m adalah konstanta)

Kunci pembangkit adalah X_0 yang disebut umpan (*seed*).

LCG mempunyai periode tidak lebih besar dari m. LCG mempunyai periode penuh ($m - 1$) jika memenuhi syarat berikut :

1. $b1$ relatif prima terhadap m.
2. $a1 - 1$ dapat dibagi dengan semua faktor prima dari m.
3. $a1 - 1$ adalah kelipatan 4 jika m adalah kelipatan 4.
4. $m >$ maks($a1$, $b1$, X_0).
5. $a1 > 0$, $b1 > 0$.

Keunggulan dari LCG adalah cepat dan hanya membutuhkan sedikit operasi bit. Akan tetapi, LCG tidak dapat digunakan untuk kriptografi karena bilangan acaknya dapat diprediksi urutan kemunculannya (Munir, 2006).

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini adalah data teks berupa pesan rahasia. Karakter-karakter dari pesan tersebut merupakan karakter yang terdapat pada *keyboard* (ASCII *printable character*). Selain data teks, data yang digunakan adalah gambar (*image*) berekstensi .bmp, .png, .gif.

3.2 Langkah-Langkah Penelitian

Langkah-langkah yang akan dilakukan pada penelitian ini, secara sistematis diuraikan sebagai berikut :

a. Studi Literatur

Tahap ini dilakukan untuk mempelajari beberapa teori terkait dengan penelitian yang dilakukan. Teori yang dipelajari dalam hal ini adalah teknik kriptografi algoritma *Affine Cipher* dan steganografi menggunakan metode *Parity Coding* pada citra digital serta *Pseudo Random Number Generator* (PRNG) untuk membangkitan bilangan acak sebagai letak penyisipan pesan.

b. Analisa Data

1) Enkripsi *plaintext* menggunakan algoritma *Affine Cipher*

Plaintext yang berbentuk teks dienkripsi menggunakan algoritma *Affine Cipher* dengan dua kunci a dan b dimana kunci a harus relatif prima dengan n yaitu jumlah karakter yang tersedia (gcd(a, n)=1). Proses enkripsi dilakukan seperti pada persamaan (2.3) sehingga dihasilkan *ciphertext*.

2) Menentukkan letak penyisipan dengan PRNG

Tiap karakter pada *ciphertext* disisipkan secara acak. Penentuan letak penyisipan dibangkitkan menggunakan PRNG metode *Linear Congruential Generator* (LCG). LCG dapat membangkitkan deretan bilangan acak hingga $m-$

1 tanpa terjadi pengulangan bilangan, dimana m merupakan modulus atau dapat juga dikatakan sebagai periode dimana bilangan acak tersebut berulang. Dengan menentukkan nilai $a1, b1$ dan m atau disebut dengan *initial value* pada (2.5) berdasarkan ketentuan yang ada, maka LCG dapat membangkitkan penuh $m - 1$ periodik sehingga dapat digunakan untuk menyisipkan tiap karakter dari *ciphertext* ke citra digital.

3) Penyisipan dengan steganografi *Parity Coding*

Setelah ditentukan letak penyisipan, selanjutnya adalah menyisipkan *ciphertext* ke dalam citra sesuai letak penyisipan yang telah dibangkitkan. Penyisipan dilakukan menggunakan metode *parity coding*, sehingga menghasilkan *stego object*. *Hiddentext* yang disisipkan ke dalam *cover object* adalah *initial value* untuk penyisipan *ciphertext*, kunci a dan b, serta jumlah karakter.

4) Enkripsi *Stego Object*

Stego object dari langkah kedua dienkripsi kembali menggunakan algoritma *Affine Cipher* dengan kunci a dan b yang sama dengan kunci pada proses enkripsi *plaintext*, sehingga selain gcd(a, n)=1, juga harus gcd($a, 256$)=1. Proses enkripsi *stego object* menggunakan persamaan (2.3) sehingga didapatkan citra kripto.

5) Dekripsi Citra Kripto

Proses ini menggunakan persamaan (2.4) dengan kunci a dan b yang sama pada proses enkripsi *stego object*. Tetapi sebelum itu, dihitung terlebih dahulu a^{-1} menggunakan persamaan (2.2) kemudian dilakukan proses dekripsi citra kripto untuk mendapatkan *stego object* semula.

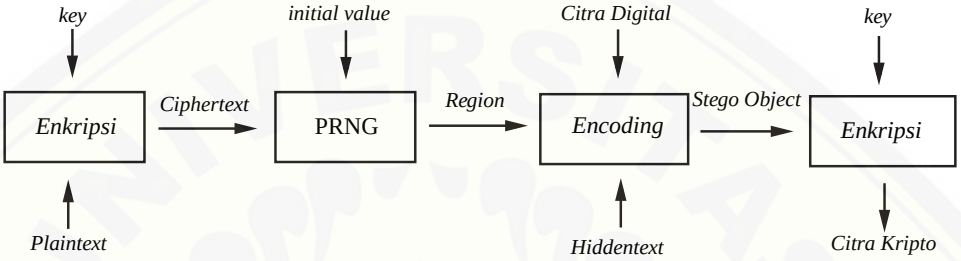
6) Ekstraksi Pesan (*Decoding*)

Pesan yang telah disisipkan diekstrak menggunakan metode *Least Significant Bit*. Hasil dari langkah ini adalah *hiddentext* yang berupa *ciphertext*, kunci a dan b, letak penyisipan karakter, serta jumlah karakter.

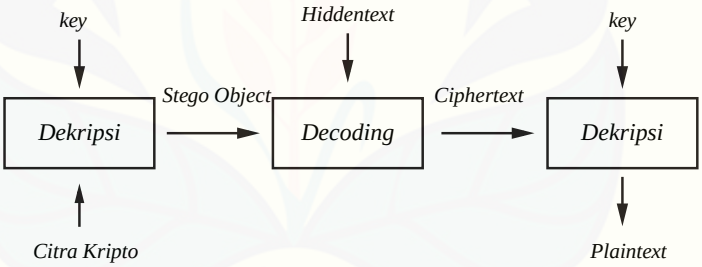
7) Dekripsi *Ciphertext* hasil Ekstraksi Pesan

Ciphertext hasil *decoding* didekripsi menggunakan persamaan (2.4). Namun sebelum itu, dihitung terlebih dahulu a^{-1} menggunakan persamaan (2.2) sehingga didapatkan *plaintext*.

dengan menggunakan diagram, maka langkah-langkah pada proses ini adalah sebagai berikut :



Gambar 3.1 Proses enkripsi *plaintext*, *encoding* pesan, dan enkripsi *stego object*



Gambar 3.2 Proses dekripsi *citra kripto*, *decoding* pesan, dan dekripsi *ciphertext*

c. Perancangan Program

Pada langkah ini menggunakan *software* MatLab 2009a dan melakukan perancangan desain GUI (*Graphic User Interface*) seperti tata letak tombol-tombol untuk tiap proses yang dibutuhkan serta tata letak *properties* pendukung program yang lainnya.

d. Pembuatan Program

Pembuatan program dilakukan berdasarkan konsep algoritma *Affine Cipher* untuk

proses enkripsi dan dekripsi data teks dan citra digital hasil steganografi dengan metode *parity coding*.

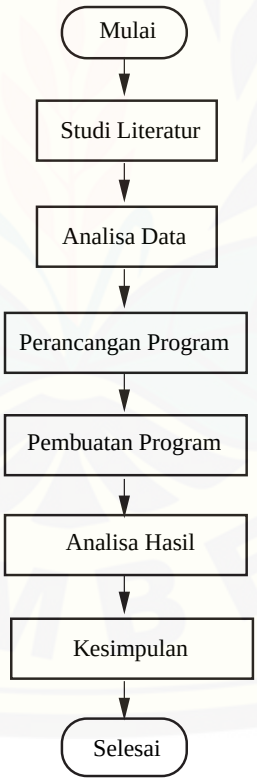
e. Analisis Hasil

Menguji hasil program agar sesuai dengan konsep metode yang digunakan, kemudian membandingkan data sebelum dan sesudah dilakukan proses.

f. Kesimpulan

Mengambil kesimpulan dari penelitian yang dilakukan, yaitu menganalisis hasil sebelum dan sesudah citra disisipkan pesan dan dilakukan proses enkripsi. Selain itu, dilakukan analisis waktu komputasi untuk proses penyisipan, enkripsi citra, dekripsi citra, serta mengambil pesan.

Berikut merupakan *flowchart* dari langkah-langkah penelitian.



Gambar 3.3 Diagram Alir Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

a. Waktu yang dibutuhkan unntuk proses penyembunyian dan pengambilan pesan semakin lama seiring bertambahnya jumlah karakter yang disisipkan.

b. Penggunaan PRNG dalam penyisipan pesan dengan metode *parity coding* cukup untuk meningkatkan keamanan penyembunyian pesan. Hal ini karena *steganaliser* harus mengetahui konstanta nilai $a1$, $b1$, dan m pada LCG untuk mengetahui posisi region dari pesan yang disembunyikan serta panjang karakter yang disisipkan.

c. Hasil enkripsi *stego object* dipengaruhi oleh pemilihan pasangan kunci. Semakin besar nilai a, maka semakin baik citra kripto yang dihasilkan (tidak terdeteksi gambar aslinya). Waktu komputasi untuk enkripsi dan dekripsi citra tidak terlalu membutuhkan waktu yang lama, tercatat waktu yang dibutuhkan adalah 0.2891 untuk proses enkripsi dan 0.298694 untuk proses dekripsi.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah:

a. Menerapkan algoritma PRNG ke metode steganografi yang lain seperti *Filtering and Masking*, dan *Algorithms and Transformation*.

b. Menggunakan algoritma PRNG yang lain seperti *Lagged Fibonacci Generator*, *Linear Feedback Shift Register* (LFSR), *Blum Blum Shub, Fortuna*, dan *Mersenne Twister*

c. Menerapkan steganografi pada media lain seperti audio atau video

DAFTAR PUSTAKA

Ariyus, D. 2006. *Kiptografi Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

Bambang, H. 2003. *Struktur Data*. Bandung: Penerbit Informatika.

Cox, I. J., Matthew L. M., Jeffry A. B., Jessica F., dan Ton K. 2008. *Digital Watermarking and Steganography Second Edition*. United States of America: Morgan Kaufan Publisher.

Herianto. 2008. *Pembangunan Perangkat Lunak Steganografi Audio MP3 dengan Teknik Parity Coding pada Perangkat Mobile Phone. Artikel Ilmiah*. Bandung: Institut Teknologi Bandung

Kromodimoeljo, S. 2009. *Teori dan Aplikasi Kiptografi*. Jakarta: SPK IT Consulting.

Munir, R. 2006. *Diktat Kuliah IF5054 Kriptografi*. Jakarta: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika.

Pramono A dan Sujjada A. 2009. *Implementasi Algoritma Hill Cipher Sebagai media Steganografi Menggunakan Metode LSB*. Bandung: Informatika.

Prijono, A. dan Wijaya, M.C. 2007. *Pengolahan Citra Digital Menggunakan MATLAB*. Bandung: Penerbit Informatika.

Purba, TB. 2012. *Implementasi Penyembunyian Dan Penyandian Pesan Pada Citra Menggunakan Algoritma Affine Cipher Dan Metode Least Significant Bit*. Tidak Diterbitkan. Skripsi. Sumatra Utara: Universitas Sumatra Utara.

Rachmawanto, E. H. 2009. *Teknik Keamanan Data Menggunakan Kriptografi dengan Algoritma Vernam Cipher dan Steganografi Metode End of File (EOF)*. Tidak Diterbitkan. Skripsi. Semarang: Universitas Dian Nuswantoro.

Roesdiana, G. T. 2015. *Pengkodean Citra Digital Hasil Steganografi Dengan Metode Least Significant Bit Untuk Data Terenkripsi dengan Algoritma Hill Cipher*. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.

Susanti, I. 2007. *Penerapan Steganografi Gambar Pada Least Significant Bit (LSB) Dengan Pengunaan PRNG (Pseudo Random Number Generator)*. Tidak Diterbitkan. Skripsi. Bogor: Institut Pertanian Bogor.

LAMPIRAN

LAMPIRAN A. Skrip Program Mencari FPB

fpb.m

```
function fp=fpb(m,n)
if n<m
    t=m;m=n;n=t;
end
while n~=0
    r=mod(m,n);
    m=n;
    n=r;
end
fp=m;
```

LAMPIRAN B. Skrip Program Menentukan invers Modulo

fpb.m

```
function gp=inverz(m,n)
cek=fpb(n,m);
if cek~=1
    error('fpb harus 1');
end
k=0;
m1=(1+k*n)/m;
r=mod(m1,2);
while r~=0 && r~=1
    k=k+1;
    m1=(1+k*n)/m;
    r=mod(m1,2);
end
gp=m1;
```

LAMPIRAN C. Skrip Program Enkripsi *Plaintext*

affine.m

```
function [buat oke]=affine(inpt,a,b)
n=length(inpt);
hrf=char(32:126);
cek=fpb(length(hrf),a);
if cek~=1
    error('fpb kunci a dengan jumlah karakter harus bernilai 1')
end
bo=0:94;
for i=1:n
    for j=1:length(hrf)
        if inpt(i)==hrf(j)
            k(i)=bo(j);
            c(i)=mod(a*k(i)+b,length(hrf));
            break
        end
    end
end
for i=1:n
    for j=1:length(hrf)
        if c(i)==bo(j)
            ciper(i)=hrf(j);
            break
        end
    end
end
oke=c;
buat=ciper;
```

LAMPIRAN D. Skrip Program Dekripsi *Ciphertext*

affine_baca.m

```
function buat=affine_baca(inpt,a,b)
n=length(inpt);
hrf=char(32:126);
a1=inverz(a,length(hrf));
bo=0:94;
for i=1:n
    for j=1:length(hrf)
        if inpt(i)==hrf(j)
            k(i)=bo(j);
            c(i)=mod(a1*(k(i)-b),length(hrf));
            break
        end
    end
end
for i=1:n
    for j=1:length(hrf)
        if c(i)==bo(j)
            ciper(i)=hrf(j);
            break
        end
    end
end
buat=ciper;
```

LAMPIRAN E. Skrip Program PRNG Algoritma LCG

prng.m

```
function r=prng(c,ack)
a=8761; b=53971; t=1;i=1; acak(1)=ack(1);
while t<c+1
    tanda=0;
    i=i+1;
    ack(i)=mod(ack(i-1)*a+b,576000);
    for j=1:i-1
        if ack(i)>ack(j)
            sel=ack(i)-ack(j);
        else
            sel=ack(j)-ack(i);
        end
        if sel<=8 || ack(i)>=(576000-8)
            tanda=1;
            break
        end
    end
    if tanda==0
        t=t+1;
        acak(t)=ack(i);
    end
end
r=acak(2:c+1);
```

LAMPIRAN F. Skrip Program Penyisipan Pesan dengan Steganografi *Parity Coding*

parity_code.m

```
function sisip=parity_code(matriks,region,karsip,rgb)
mat=matriks(:,:,rgb);
t=region;
for i=1:length(karsip)
    binkar=digit2biner(karsip(i));
    for j=1:8
        a=digit2biner(double(mat(t)));
        p=a(1);
        for k=2:8
            p=xor(p,a(k));
        end
        if binkar(j)~=p
            if (p==0 && a(8)==0) || (p==1 && a(8)==1)
                a(8)=binkar(j);
            else
                a(8)=p;
            end
            mat(t)=biner2digit(a);
        else
            mat(t)=biner2digit(a);
        end
        t=t+1;
    end
end
sisip=mat;
```

LAMPIRAN G. Skrip Program Enkripsi Citra dengan *Affine Cipher*

affine_citra.m

```
function gbr2=affine_citra(gbr2,x,y)
cek=fpb(256,x);
if cek~=1
    error('fpb kunci a dengan jumlah karakter harus bernilai 1')
end
[m n o]=size(gbr2);
for i=1:m
    for j=1:n
        b(j)=double(gbr2(i,j,1));
        c(j)=double(gbr2(i,j,2));
        d(j)=double(gbr2(i,j,3));
    end
    gbr2(i,:,1)=mod(x*b+y,256);
    gbr2(i,:,2)=mod(x*c+y,256);
    gbr2(i,:,3)=mod(x*d+y,256);
end
```

LAMPIRAN H. Skrip Program Dekripsi Citra dengan *Affine Cipher*

affinebaca_citra.m

```
function gbr3=affinebaca_citra(gbr3,x1,y)
cek=fpb(256,x1);
if cek~=1
    error('fpb kunci a dengan jumlah karakter harus bernilai 1')
end
[m n o]=size(gbr3);
x=inverz(x1,256);
for i=1:m
    for j=1:n
        b(j)=double(gbr3(i,j,1));
        c(j)=double(gbr3(i,j,2));
        d(j)=double(gbr3(i,j,3));
    end
    gbr3(i,:,1)=mod(x*(b-y),256);
    gbr3(i,:,2)=mod(x*(c-y),256);
    gbr3(i,:,3)=mod(x*(d-y),256);
end
```

LAMPIRAN I. Skrip Program Pengambilan Pesan dengan Steganografi *Parity Coding*

parity_code_ekstrak.m

```
function decod=parity_code_ekstrak(matr,reg,rgb,jmlkar)
mat=matr(:,:,rgb);
p=0; d=0;t=reg;
for i=1:8*jmlkar
    A=digit2biner(double(mat(t)));
    k=A(1);
    for j=2:8
        k=xor(k,A(j));
    end
    p=p+1;
    a(p)=k;
    if p==8
        p=0;
        d=d+1;
        decod(d)=biner2digit(a);
    end
    t=t+1;
end
```

LAMPIRAN J. Skrip Program Konversi Desimal ke Biner

digit2biner.m

```
function biner=digit2biner(m)
p=zeros(1,8);
if m==0
    biner=p;
    return
end
i=0;
while m~=1
    a=mod(m,2);
    i=i+1;
    if a==0
        p(i)=0;
        m=m/2;
    else
        p(i)=1;
        m=(m-a)/2;
    end
end
p(i+1)=1;
t=length(p);
q(1,1:t)=p(1,t:-1:1);
biner=q;
```

LAMPIRAN K. Skrip Program Konversi Biner ke Desimal

biner2digit.m

```
function biner=biner2digit(m)
n=length(m);
jml=0;
for k=1:n
    a(k)=2^(n-k);
    jml=jml+m(k)*a(k);
end
biner=jml;
```