ISSN 2085-4552

Kriptografi Citra Menggunakan Metode Rivest-Shamir-Adleman Chinese Remainder Theorem Di Konsultan XYZ

Eko Budi Setiawan, Yogie Setiawan Nugraha
Program Studi Teknik Informatika, Universitas Komputer Indonesia, Bandung, Indonesia
ekobudisetiawan@ymail.com, yogie_setiawan_nugraha@ymail.com

Diterima 27 Agustus 2015
Disetujui 2 Oktober 2015

***Abstract*— This study basically do image management processes which have no restrictions in about permissions to view the images on the XYZ consultant. This results in image can be easily used by those who do not have the right to open it. Therefore, security research digital image with cryptographic techniques using Rivest Shamir-Adleman With the Chinese Remainder Theorem (RSA-CRT) aims to provide restrictions permissions to view the image, and give the number of restrictions to view images that can be performed by the client. Results of this research is a security application that can create digital images can not be seen without using a specific key and restrictions as well as providing an image to only be opened in a few times.**

***Index Terms*— Cryptography, Digital Image, RSA-CRT, Data Security, Asymmetric Key**

I. PENDAHULUAN

Konsultan XYZ adalah salah satu konsultan *art* dan *design* di Bandung. Perusahaan ini menangani beberapa pekerjaan yang berhubungan dengan desain seperti *branding,* perencanaan dan desain bangunan serta penelitian dan hiburan. Berdasarkan informasi yang didapat dari pihak Konsultan XYZ diketahui bahwa ada klien yang tidak melanjutkan kerjasama dengan mereka tetapi tetap menggunakan desain dari Konsultan XYZ. Jika kasus tersebut terus terjadi maka akan mengakibatkan kerugian besar untuk pihak Konsultan XYZ mengingat dalam dunia desain, konsep suatu desain itu sangat berharga. Meskipun pada implementasinya suatu desain mengalami perubahan namun dengan konsep yang sama bisa dikatakan tetap menggunakan konsep dari pihak Konsultan XYZ.

Berdasarkan kasus yang ada maka diperlukan sebuah aplikasi untuk mengamankan gambar di perusahaan Konsultan XYZ tersebut. Dalam hal ini khususnya untuk membatasi klien untuk melihat desain agar mengurangi penggunaan tanpa izin oleh klien. Salah satu caranya adalah dengan cara menyamarkan gambar digital sehingga tidak bisa dilihat tanpa menggunakan kode dari pihak perusahaan.

Penelitian kali ini akan menggunakan pengamanan dengan teknik kriptografi yang dapat menyamarkan gambar digital menggunakan metode *Rivest-Shamir-Adleman* dengan *Chinese Remainder Theorem* (RSA-CRT). Metode RSA-CRT dipilih karena berdasarkan hasil penelitian Rini Wati Lumbangaol [1] diketahui bahwa RSA bisa digunakan untuk enkripsi gambar. Namun berdasarkan buku Kriptografi Untuk Keamanan Jaringan [2] dikatakan bahwa metode RSA membutuhkan waktu lebih lama dalam proses dekripsi sehingga ditambahkan teorema CRT untuk mempercepat proses dekripsi tersebut.

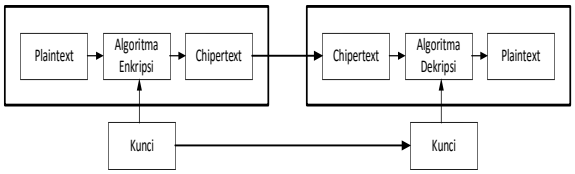
Dengan demikian maka penelitian ini akan berfokus untuk membangun aplikasi keamanan yang yang bert untuk akan mengurangi kasus yang terjadi. Selain itu juga dengan teknik kriptografi ini juga dapat mengurangi kemungkinan pihak selain klien dan perusahaan untuk melihat desain yang dibuat.

II. PENJELASAN ALGORITMA

Kriptografi modern tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [2]. Secara umum kriptografi

dapat dibedakan menjadi dua berdasarkan era nya, yaitu kriptografi klasik yang mana ada sebelum era komputer dengan menggunakan kunci simetrik subtitusi atau transposisi, serta kriptografi modern yang ada setelah era komputer yang menggunakan teknik penyandian lebih bervariasi.

Secara umum sistem penyandian dapat digambarkan dengan lebih sederhana yaitu seperti berikut:



Gambar 1. Gambaran Sistem Kriptografi

Berdasarkan gambar 1 tersebut terdapat beberapa istilah sebagai berikut :

a. *Plaintext*

Plaintext meruapakan teks asli yang dapat terbaca. *Plaintext* merupakan masukan bagi algoritma enkripsi.

b. Kunci

Kunci merupakan masukan bagi algoritma enkripsi juga selain *plaintext.* Kunci dapat dibedakan menjadi tiga yaitu :

1) Kunci simetrik

Kunci simetrik merupakan kunci yang proses enkripsi dan dekripsi menggunakan kunci yang bernilai sama.

2) Kunci asimetrik

Kunci asimetrik merupakan kunci yang proses enkripsi dan dekripsinya menggunakan kunci yang bernilai beda.

3) Fungsi Hash

Fungsi *hash* merupakan fungsi yang melakukan pemetaan pesan dengan panjang sembarang ke sebuah teks khusus yang disebut *message digest* dengan panjang yang tetap. Fungsi *hash* umumnya dipakai sebagai nilai uji (*check value)* pada mekanisme keutuhan data.

c. Algoritma enkripsi

Algoritma enkripsi memiliki dua masukan yaitu teks asli (*plaintext)* dan kunci rahasia. Algoritma ini melakukan transformasi terhadap *plaintext* dengan menggunakan kunci yang ada, algoritma ini akan menghasilkan teks ter-sandi (*chipertext)*.

d. Algoritma dekripsi

Algoritma ini memiliki dua masukan yaitu *chipertext* dan kunci rahasia. Algoritma ini memulihkan kembali *chipertext* menjadi teks asli jika kunci yang digunakan untuk melakukan dekripsi bernilai benar.

e. *Chipertext*

Chipertext adalah hasil *output* dari algoritma enkripsi. *Chipertext* dapat dikatakan sebagai pesan dalam bentuk tersembunyi. Algoritma enkripsi yang baik akan menghasilkan *chipertext* yang terlihat acak.

Algoritma enkripsi dan dekripsi sistem kriptografi RSA berasumsi pada fungsi satu arah (*one – way function*) yang dibangun oleh fungsi eksponensial modular pada grup perkalian (Z^*_n, x) dan grup perkalian ($Z^*_{\phi(n)}$, x) dengan n = p x q, p, q adalah bilangan prima dan $\phi(n) = (p - 1) \times (q - 1)$.

Sedangkan metode RSA – CRT (Rhivest – Shamir – Adleman dengan Chinese Remainder Theorem) merupakan suatu metode kriptografi yang sama dengan RSA biasa, namun memanfaatkan teorema CRT untuk memperpendek ukuran bit eksponen dekripsi *d* dengan cara menyembunyikan *d* pada sistem yang kongruen sehingga mempercepat waktu dekripsi. Sistem yang kongruen pada CRT adalah dengan pemanfaatan penyelesaian masalah kongruen dengan modulus berbeda. Berikut penggambaran sistem kongruen pada CRT :

$$x \equiv a_1 \bmod m_1$$

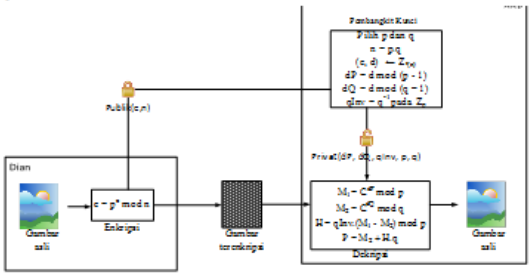
$$x \equiv a_2 \bmod m_2$$

...

$$x \equiv a_k \bmod m_k$$

Secara sederhana penggunaan CRT pada nilai *d* mengakibatkan pemecahan kunci sehingga menghasilkan kunci baru yaitu parameter dP, dQ dan qInv yang memiliki ukuran setengah panjang bit *d.* Kunci private RSA – CRT ditetapkan sebagai $K_{private}$ = (dP, dQ, qInv, p, q) [11].

Untuk gambaran sistem terlihat seperti berikut :

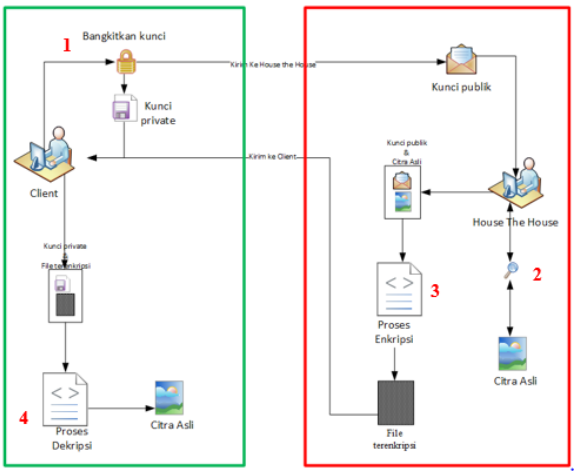


Gambar 2. Gambaran Umum Sistem Kriptografi RSA – CRT

III. PEMBAHASAN PENELITIAN

A. Analisis Aplikasi yang Akan Dibangun

Proses yang akan terjadi dalam aplikasi yang akan dibangun adalah sebagai berikut :

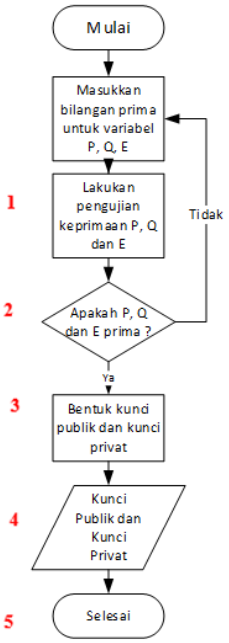


Gambar 3. Proses kriptografi terhadap citra

A.1 *Analisis Pembangkit Kunci*

Proses pembangkit kunci dilakukan terlebih dahulu oleh klien. Proses ini akan menghasilkan kunci *private* yang digunakan untuk proses dekripsi dan kunci publik yang akan digunakan pihak perusahaan untuk proses enkripsi. Proses ini mewakili model proses kriptografi pada gambar 3 pada tahapan 1.

Berikut flowchat yang akan menggambarkan proses pembangkit kunci dari sebuah metode RSA – CRT :



Gambar 4. Flowchart Pembangkit Kunci

A.2 Analisis Pembangkit Kunci

Flowchart pada gambar 4 proses 1, 4 dan 5 dijelaskan dalam bentuk algoritma pada tabel 1 berikut :

Tabel 1. Algoritma Pembangkit Kunci RSA - CRT

Tahap	**Algoritma** Pembangkit Kunci RSA – CRT
1	Bangkitkan bilangan prima besar p dan q.
2	n ← p * q
3	ϕ(n) ← (p – 1) * (q – 1)
4	e ← $Z_{\phi(n)}$ dengan gcd(e, ϕ(n)) = 1
5	d ← e^{-1} pada $Z_{\phi(n)}$
6	dP ← d mod (p – 1)
7	dQ ← d mod (q – 1)
8	qInv = q^{-1} pada Z_p K_{publik} = (e, n), $K_{private}$ = (dP, dQ, qInv, p, q)

Misal dari proses pembangkit kunci didapatkan :

Kunci publik : (73, 527)

Kunci private : (9, 7, 11, 17, 31)

A.3 Analisis Citra Digital

Sebelum proses enkripsi dilakukan maka pihak konsultan XYZ harus menyediakan terlebih dahulu citra yang akan dilakukan enkripsi. Proses ini merupakan penggambaran proses yang terjadi

pada gambar 3 tahap 2.

Citra ini dapat berformat .jpg dan .bmp dengan pembatasan ukuran yaitu ukuran terbesar lebar dan tinggi yaitu 1037 x 384 piksel. Berikut merupakan proses pengolahan citra sehingga citra siap digunakan pada proses dekripsi :

Gambar 5. Flowchat pengambilan nilai citra digital

Sebagai gambaran akan digunakan sebuah citra dengan ukuran 360 x 273 pixel seperti berikut:

Gambar 6. Citra digital ukuran 360 x 273

Dengan menggunakan fungsi dalam bahasa C# yaitu *Image.Save(Stream, Format)* didapat nilai dari citra digital tersebut yaitu 53292. Dari nilai 53292 yang didapatkan akan digunakan 10 data terakhir dari citra untuk menggambarkan proses enkripsi dan dekripsi. Berikut adalah nilai yang diambil :

Array ke →	53282	69
		20
		80
		1
		69
		20
		80
		7
		255
	217	53292 ← array ke

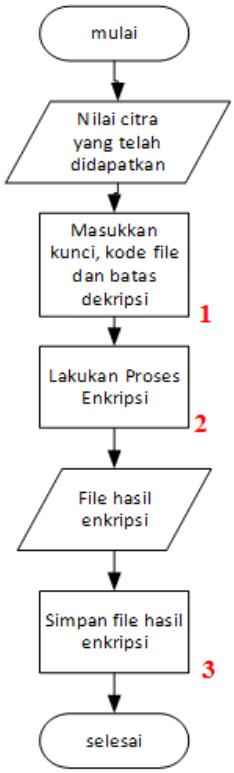
Gambar 7. Nilai citra digital dari gambar 6

A.4 Analisis Proses Enkripsi

Proses enkripsi ini mewakili proses ke 3 pada gambar 3. Proses ini dilakukan oleh pihak Konsultan XYZ untuk menyandikan gambar sehingga gambar tidak bisa dibuka dengan aplikasi pembuka citra digital.

Proses yang dilakukan untuk melakukan enkripsi dengan metode RSA–CRT ini sama dengan melakukan enkripsi pada RSA biasa. Hal ini karena teorema CRT hanya digunakan pada proses dekripsi yang juga berakibat pada proses pembangkit kunci.

Secara umum proses enkripsi yang terjadi dapat terlihat seperti pada flowchart berikut ini :



Gambar 8. Flowchart proses enkripsi

Berdasarkan gambar 8, proses 1 dan 2 dijelaskan pada algoritma enkripsi RSA–CRT. Sedangkan untuk proses 3 lebih dikhususkan pada tahapan implementasi karena proses penyimpanan hanya bisa dilakukan melalui proses komputasi. Untuk algortima pada proses enkripsi cukup sederhana. Hal ini bisa terlihat pada tabel 2 berikut :

Tabel 2. Algoritma Enkripsi RSA - CRT

	Algoritma Enkripsi RSA - CRT
1	**Input** : $K_{publik} = (e, n)$, $P \in Z_n$
2	**Output** : C □ $Z_{.}$
3	$C = P^e \bmod n$

Dari algoritma diatas terlihat bahwa P merupakan masukan untuk dilakukan proses enkripsi. Dalam kasus kriptografi pada citra ini nilai P merupakan masukan berupa nilai *byte array* yang didapatkan dari suatu file citra.

Sebagai contoh di ambil gambar berekstensi .jpg berukuran 360 x 273 sebagai berikut :

Gambar 9. Citra jpg 360 x 273 yang akan dienkripsi

Selanjutnya adalah diambil nilai dari citra digital tersebut yang akan ditampilkan dalam bentuk sampel yang digambarkan dalam bentuk array satu dimensi. Dalam hal ini akan diambil 10 nilai terakhir dari citra tersebut :

Array ke →	53282	69
		20
		80
		1
		69
		20
		80
		7
		255
217	53292	← Array ke

Gambar 10. Nilai terakhir dari gambar 6

Selanjutnya nilai tersebut yang akan dilakukan proses enkripsi. Dari proses pembentukan kunci telah didapatkan nilai *e* dan *n* yang merupakan bahan untuk melakukan enkripsi. Nilai e = 73 dan nilai n = 527 selanjutnya dimasukkan kedalam algoritma kriptografi dengan sample nilai dari tabel 2.

Dalam sistem enkripsi RSA dilakukan dengan menggunakan sistem eksponensial modular ($C = P^e \bmod n$). Pada persamaan tersebut memungkinkan proses yang memakan waktu dan memori yang besar ketika nilai *P* yang dipangkatkan dengan nilai yang *e* besar. Untuk mempercepat proses pada sistem eksponensial modular tersebut dilakukan dengan menggunakan algoritma *square and multiply* [2].

Dengan menggunakan algoritma enkripsi dan memakai prinsip eksponensial modular

dalam proses perhitungannya didapatkan nilai citra yang terenkripsi sebagai berikut :

Array ke →	53282	205
		320
		175
		1
		205
		320
		175
		112
		391
	53292	217 ← Array ke

Gambar 11. Nilai citra yang terenkripsi

Dari nilai diatas sangat terlihat perubahan nilai yang sangat signifikan yang melampui batas tipe data *byte* yaitu 255. Sehingga jika akan melakukan proses enkripsi sebaiknya mengubah tipe data menjadi tipe data *long*. Selain itu juga seperti yang telah dibahas pada analisis citra digital bahwa dengan nilai yang melebihi 255 tidak akan bisa melihat citra tetapi yang kita bisa lihat hanya sebuah file yang berisi nilai saja, serta file tersebut akan diberi ekstensi .jpg.

Pada proses ini pula akan ditambahkan dua buah indeks array untuk menampung kode file dan batas dekripsi dari citra terenkripsi tersebut. Kedua nilai tersebut digunakan untuk membatasi proses dekripsi sehingga proses dekripsi hanya bisa dilakukan sebatas yang diinginkan pihak perusahaan yang dalam hal ini yaitu Konsultan XYZ.

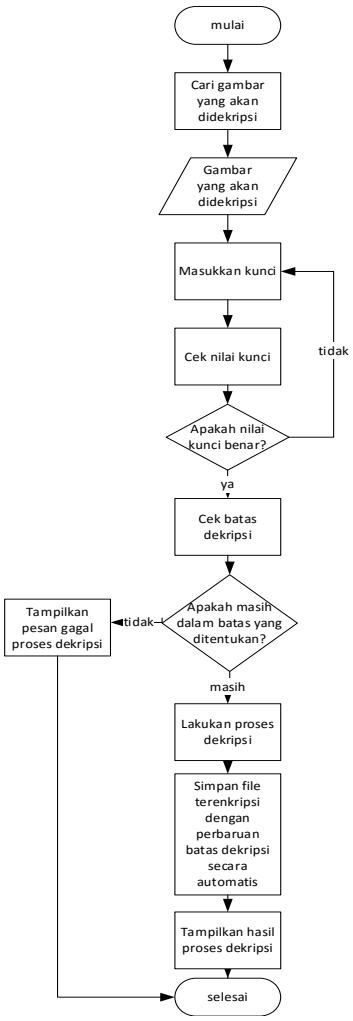
Misalkan bahwa dengan kode file HTH dan batas 2 kali dekripsi akan kita masukkan ke dalam array yang baru ditambahkan, maka array baru yang terbentuk adalah sebagai berikut :

Array ke →	53282	205	
		320	
		175	
		1	
		205	
		320	
		175	
		112	
		391	
		217	53292 ← Array ke
Penyisipan kode file	HTH →	HTH	53293
Penyisipan pembatas dekripsi	2 →	2	53294

Gambar 12. Nilai citra terenkripsi setelah penyisipan kode file dan batas dekripsi

A.5 Analisis Proses Dekripsi

Proses dekripsi merupakan proses membalikan nilai yang sudah terenkripsi dari suatu *chipertext* ke bentuk semula dalam bentuk *plaintext* dalam kasus ini akan diubah ke citra semula. Alur proses dekripsi dapat terlihat pada flowchart berikut ini :



Gambar 13. Flowchart proses dekripsi

Adapun algoritma dari proses dekripsi dapat terlihat sebagai berikut :

Tabel 3 Algoritma Proses Dekripsi RSA - CRT

	Algoritma Dekripsi
	Input : C = P^e mod n, K_{private} = (dP, dQ
	Output : P
1	$m_1 = C^{dP} \bmod p$
2	$m_2 = C^{dQ} \bmod q$
3	$h = qInv\ (m_1 - m_2) \bmod p$
4	$P = m_2 + h.q$

Berikut merupakan gambaran dari proses dekripsi yang akan dilakukan :

Array ke →	53282	205
		320
		175
		1
		205
		320
		175
		112
		391
	217	53292 ←Array ke
	HTH	53293
	2	53294

Gambar 14. Citra terenkripsi yang akan dilakukan proses dekripsi

$K_{private} = (9, 7, 11, 17, 31)$

Dengan nilai P = 205 pada larik ke 53282 maka akan terlihat proses berikut

a. $m_1 = C^{dP} \bmod p = 205^9 \bmod 17 = 1$

b. $m_2 = C^{dQ} \bmod q = 205^7 \bmod 31 = 7$

c. $h = qInv * (m_1 - m_2) \bmod p = 11 * (1 - 7) \bmod 17 = 2$

d. $P = m_2 + (h * q) = 7 + ((2) * 31) = 69$

Terlihat bahwa nilai P hasil dekripsi adalah 69 dan sesuai nilai asli citra pada array citra asli pada larik 53282 (gambar 14). Lakukan hal yang sama pada nilai citra terenkripsi lainnya. Sehingga menghasilkan nilai hasil dekripsi sebagai berikut :

Array ke →	53282	69
		20
		80
		1
		69
		20
		80
		7
		255
	217	53292 ←Array ke

Gambar 15. Nilai citra setelah proses dekripsi

Pada proses ini juga mengurangi nilai batas dekripsi sehingga nilai citra terenkripsi yang baru sebagai berikut :

Array ke →	53282	205
		320
		175
		1
		205
		320
		175
		112
		391
	217	53292 ← Array ke
	HTH	53293
Penyisipan pembatas dekripsi yang baru	2 – 1 →	1 53294

Gambar 16. Nilai citra terenkripsi yang baru

IV. Implementasi Aplikasi

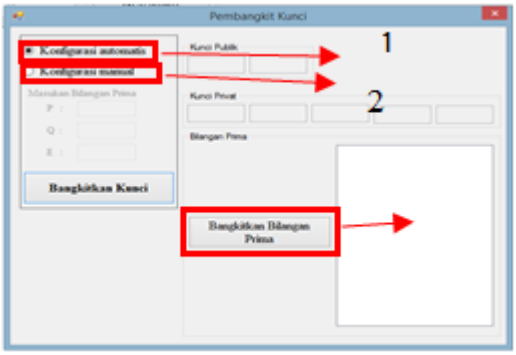
Aplikasi kriptografi pada citra digital ini merupakan aplikasi yang dapat digunakan secara langsung tanpa melalui proses instalasi. File *executable* yang ada dapat langsung digunakan.

A. Implementasi Antarmuka

Antarmuka dari aplikasi ini terdiri dari dua bagian yaitu antramuka enkripsi dan antarmuka dekripsi. Antarmuka dekripsi memiliki satu antarmuka turunan yaitu antarmuka pembangkit kunci. Berikut rincian dari implementasi antarmuka yang dibuat:

A.1 Implementasi Antarmuka Pembangkit Kunci

Antarmuka pembangkit kunci sebagai berikut:



Gambar 16. Antarmuka pembangkit kunci

Pembangkit kunci ini merupakan hasil implementasi dari algoritma pembangkit kunci RSA – CRT. Dari tampilan pada gambar 16 diatas terlihat bahwa ada dua metode pembangkit kunci yaitu sebagai berikut :

1. Konfigurasi otomatis

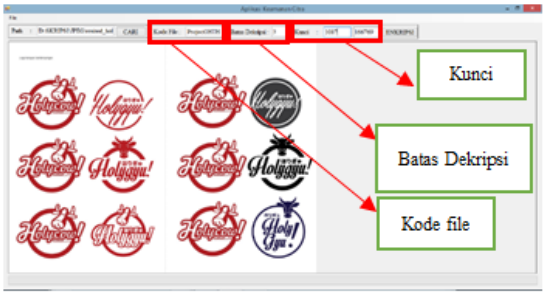
Konfigurasi otomatis ini akan membangkitkan nilai P, Q dan E secara acak sehingga pengguna tidak perlu memasukan nilai P, Q dan E.

2. Pembangkit bilangan prima

Pembangkit merupakan fasilitas yang digunakan untuk membantu pengguna dalam membangkitkan bilangan prima dikarenakan tidak semua pengguna tahu tentang bilangan prima.

A.2 Implementasi Antarmuka Enkripsi

Setelah melakukan pembangkitan kunci maka Konsultan XYZ melakukan pemilihan gambar yang akan dilakukan dan hasilnya akan dipilih gambar berikut sebagai bahan untuk dilakukan enkripsi :

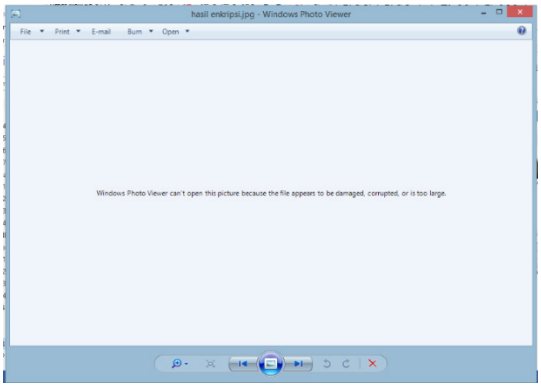


Gambar 17. Antarmuka Enkripsi

Langkah selanjutnya adalah memasukan semua isian dari tiap kolom dekripsi sebagai berikut :

1. Kode File
   Kode file dapat berupa isian apapun yang merepresentasikan file yang akan dienkripsi dan harus secara unik. Unik berarti setiap gambar yang akan dienkripsi harus memiliki kode yang berbeda.
2. Batas Dekripsi
   Batas Dekripsi digunakan untuk menentukan berapa kali orang lain dapat melakukan dekripsi terhadap file citra terenkripsi tersebut.
3. Kunci
   Kunci berisi kunci publik yang telah dibangkitkan sebelumnya.

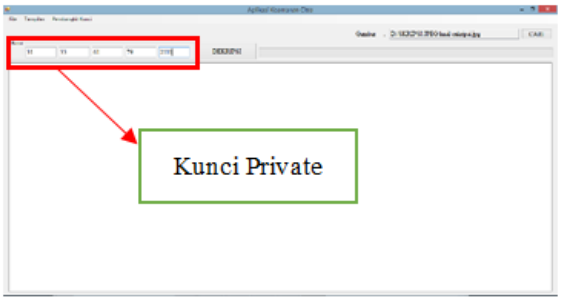
Adapun hasil enkripsi gambar yang tidak bisa dibuka adalah sebagai berikut :



Gambar 18. Hasil Enkripsi Gambar

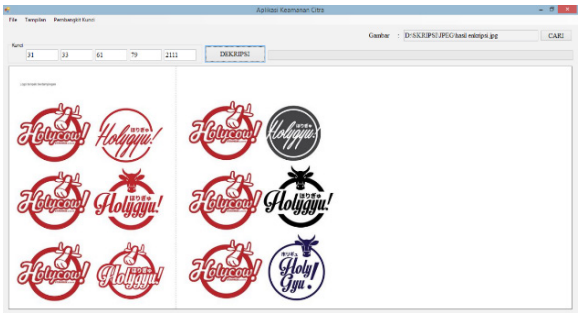
A.3 Implementasi Antarmuka Dekripsi

Setelah dilakukan enkripsi maka gambar akan dikirimkan kepada klien. Untuk melihat gambar, klien harus memiliki aplikasi dekripsi dan kunci private. Selanjutnya adalah mencari gambar yang akan dilakukan dan akan mengisi kunci private yang ada pada form dekripsi sebagai berikut :



Gambar 19. Antarmuka Dekripsi

Setelah memasukan kunci private maka akan dilakukan proses dekripsi pada gambar yang dipilih dan hasilnya adalah sebagai berikut :

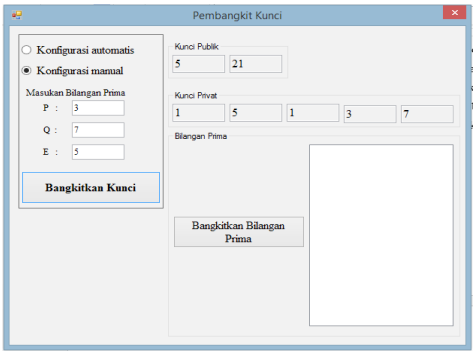


Gambar 20. Hasil Dekripsi Gambar

A.4 Pengujian Penyerangan Sistem

Untuk menguji penyerangan terhadap sistem enkripsi gambar, dilakukan menggunakan teknik brute force dimana akan melakukan percobaan terhadap semua kemungkiann yang ada atas segala data inputan. Sebagai contoh sederhana telah dibangkitkan kunci dengan komponen pengangkit terkecil yaitu P=3, Q=7, E=5.

Dari aplikasi pembangkit kunci yang ada, maka didapatkan hasil sebagai berikut :



Gambar 21. Hasil Pembangkit Kunci Sederhana

Didapatkan bahwa kunci publik = 5 – 21, serta kunci private = 1 – 5 – 1 – 3 – 7. Terlihat bahwa angka terbesar yang berhasil dibangkitkan adalah 7 yang masih dalam range 10 angka awal. Jika saja seseorang mencoba memecahkan kunci ini dengan range 10 angka awal maka didapatkan kemungkinan sebanyak 10 x 10 x 10 x 10 x 10 = 100.000 kali percobaan.

Dari percobaan tersebut apabila dilakukan dengan mengurutkan angka, maka kunci akan didapatkan pada percobaan ke 15.137 atau 15.137/100.000 percobaan. Angka yang besar ini akan sulit ditemukan terutama jika menggunakan percobaan secara manual. Selain itu, kemungkinan ini akan terus bertambah seiring dengan komponen pembangkit kunci (P,Q, dan E) yang nilainya semakin besar.

V. SIMPULAN

Berdasarkan dari hasil implementasi aplikasi dan metode kriptografi yang digunakan, didapatkan bahwa aplikasi yang dibangun telah memberikan hak akses untuk melihat gambar dengan cara menggunakan kunci yang cukup tahan terhadap serangan pihak yang mencoba membuka kunci tersebut sehingga gambar tetap aman dan tidak akan bisa dilihat oleh pihak yang tidak berkepentingan.

Berdasarkan pengujian penyerangan menggunakan teknik brute force, jika seseorang mencoba memecahkan kunci ini dengan range 10 angka awal maka akan didapatkan kemungkinan sebanyak = 100.000 kali percobaan.

Selain itu gambar yang diamankan telah memiliki nilai pembatas untuk melihat gambar tersebut sehingga dapat mengurangi penyalahgunaan oleh pihak yang memiliki hak akses yang dalam hal ini adalah klien dari pihak perusahaan *Konsultan XYZ.*

DAFTAR PUSTAKA

[1] R. W. Lumbangaol, “Aplikasi Pengamanan Gambar Dengan Algoritma Rivest-Shamir-Adleman (RSA),” STIMIK Budidarma Medan.

[2] R. Sadikin, Kriptografi Untuk Keamanan Jaringan, Yogyakarta: Andi, 2012.

[3] L. J. Meleong, Metode Penelitian Kualitatif, Bandung: Rosda, 2013.

[4] R. Indrawan dan P. Yaniawati, Metode Penelitian, Bandung: Refika Aditama, 2014.

[5] R. S. Pressman, Rekayasa Perangkat Lunak Pendekatan Praktis (Buku Satu), Yogyakarta: Andi, 2002.

[6] I. Sommerville, Software and Enginering (Rekayasa Perangkat Lunak) Edisi 6 Jilid 1, Jakarta: Erlangga, 2003.

[7] R. Munir, Pengolahan Citra digital, Bandung: Informatika.

[8] A. Kadir dan A. Susanto, Teori dan Aplikasi Pengolahan Citra, Yogyakarta: Andi, 2012.

[9] U. Ahmad, Pengolahan Citra Digital dan Teknik Pemrogramannya, Yogyakarta: Graha Ilmu, 2005.

[10] R. R. R., A. Shamir dan A. L., A Method for Obtaining Digital Signature and Public-Key Cryptosystem., Commun: ACM, 1983.

[11] A. Nugroho, Algoritma dan Struktur Data Dalam Bahas Java, Jogyakarta: Andi, 2008.

[12] B. Hariyanto, Rekayasa Sistem Berbasis Objek, Bandung: Informatika, 2004.

[13] C. Petzold, Net Book Zero : What the C or C++ Programmer Needs to Know About C# and the .Net Framework, www.charlespetzold.com, 2007.

[14] R. Faraz, T. Nestinius, J. Warthington dan L. A. Wright, C# School, www.Programmersheaven.com, 2006.

[15] A. Troelsen, Pro C# 2008 and the .Net 3.5 Platform, New York: Springer-Verlag New York, 2007.

[16] A. Nugroho, Algoritma dan Pemrograman Menggunakan Bahasa C#, Yogyakarta: Andi, 2011.