ANALISIS KEMAMPUAN ALGORITMA ELGAMAL UNTUK KRIPTOGRAFI CITRA

Yo'el Pieter Sumihar*[1]
[1,2,3] Jurusan Komputer, Teknik Informatika, Fakultas Sains dan Komputer, Universitas Kristen Immanuel
Jalan Solo Km. 11 PO Box 4 YKAP Yogyakarta, ph: (0274) 496256-296247 fax: (0274) 496258
e-mail: *[1]pieter.haro@gmail.com

Abstrak

Algoritma ElGamal merupakan algoritma kriptografi asimetris yang menggunakan dua jenis kunci, yaitu kunci publik dan kunci rahasia. Tingkat keamanan algoritma ini didasarkan atas masalah bilangan bulat modulo prima. Algoritma ElGamal mempunyai kunci publik berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini melakukan proses enkripsi dan dekripsi pada blok-blok plainteks dan dihasilkan blok-blok cipherteks yang masing-masing terdiri dari dua pasang bilangan.

Algoritma ElGamal yang digunakan dalam proses enkripsi dan dekripsi pada citra. Kemudian dibuat sebuah program pengamanan pesan rahasia berdasarkan algoritma ElGamal.

Algoritma kriptografi ElGamal dapat diimplementasikan terhadap file citra. Hasil enkripsi citra input sangat berbeda dari citra inputnya, baik dari segi tampilan dan ukuran/resolusi. Hasil enkripsi citra input selalu memiliki ukuran minimal 2 kali lebih besar dari citra aslinya. Waktu pemrosesan enkripsi dipengaruhi oleh blocksize dan resolusi, semakin kecil nilai blocksize dan semakin besar resolusi citra yang diproses maka waktu proses juga semakin lama. Citra terenkripsi dapat kembali ke citra aslinya setelah mengalami proses dekripsi. Waktu pemrosesan dekripsi dipengaruhi oleh blocksize dan resolusi, semakin kecil nilai blocksize dan semakin besar resolusi citra yang diproses maka waktu proses juga semakin lama.

***Kata kunci** : algoritma, asimetris, cipher blok, ElGamal, kriptografi, kunci publik, kunci private, citra, enkripsi, dekripsi.*

1. PENDAHULUAN

Kemajuan dan perkembangan teknologi informasi dewasa ini telah berpengaruh pada hampir semua aspek kehidupan manusia, tak terkecuali dalam hal berkomunikasi. Dengan adanya internet, komunikasi jarak jauh dapat dilakukan dengan cepat dan murah. Namun di sisi lain, ternyata internet tidak terlalu aman karena merupakan media komunikasi umum yang dapat digunakan oleh siapapun sehingga sangat rawan terhadap penyadapan informasi oleh pihak-pihak yang tidak berhak mengetahui informasi tersebut. Oleh karena penggunaan internet yang sangat luas seperti pada bisnis, perdagangan, bank, industri dan pemerintahan yang umumnya mengandung informasi yang bersifat rahasia maka keamanan informasi menjadi faktor utama yang harus dipenuhi. Berbagai hal telah dilakukan untuk mendapatkan jaminan keamanan informasi rahasia ini. Salah satu cara yang digunakan adalah dengan menyandikan isi informasi menjadi suatu kode-kode yang tidak dimengerti sehingga apabila disadap maka akan kesulitan untuk mengetahui isi informasi yang sebenarnya.

Metode penyandian yang pertama kali dibuat masih menggunakan metode algoritma rahasia. Metode ini menumpukan keamanannya pada kerahasian algoritma yang digunakan. Namun metode ini tidak efisien saat digunakan untuk berkomunikasi dengan banyak orang. Oleh karena itu seseorang harus membuat algoritma baru apabila akan bertukar informasi rahasia dengan orang lain.

Karena penggunaannya yang tidak efisien maka algoritma rahasia mulai ditinggalkan dan dikenalkan suatu metode baru yang disebut dengan algoritma kunci. Metode ini tidak menumpukan keamanan pada algoritmanya, tetapi pada kerahasian kunci yang digunakan pada proses penyandian. Algoritmanya dapat diketahui, digunakan dan dipelajari oleh siapapun. Metode algoritma kunci mempunyai tingkat efisiensi dan keamanan yang lebih baik dibandingkan dengan

algoritma rahasia. Sampai sekarang algoritma kunci masih digunakan secara luas di internet dan terus dikembangkan untuk mendapatkan keamanan yang lebih baik.

Algoritma ElGamal merupakan salah satu dari algoritma kunci. Algoritma ini dikembangkan pertama kali oleh Taher ElGamal pada tahun 1985. Sampai saat ini, algoritma ElGamal masih dipercaya sebagai metode penyandian, seperti aplikasi PGP dan GnuPG yang dapat digunakan untuk pengamanan e-mail dan tanda tangan digital. Pada tahun 1994 pemerintah Amerika Serikat mengadopsi Digital Signature Standard, sebuah mekanisme penyandian yang berdasar pada algoritma ElGamal.

2. METODE PENELITIAN

2.1 Kriptografi ElGamal

Algoritma ElGamal merupakan algoritma kriptografi asimetris. Pertama kali dipublikasikan oleh Taher ElGamal pada tahun 1985. Algoritma ini didasarkan atas masalah logaritma diskret pada grup ℤP*. Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Algoritma ini merupakan cipher blok, yaitu melakukan proses enkripsi pada blok-blok plainteks dan menghasilkan blok-blok cipherteks yang kemudian dilakukan proses dekripsi, dan hasilnya digabungkan kembali menjadi pesan yang utuh dan dapat dimengerti. Untuk membentuk sistem kriptografi ElGamal, dibutuhkan bilangan prima p dan elemen primitif grup ℤP*.

Untuk lebih jelasnya mengenai algoritma ElGamal, berikut ini diberikan suatu sistem kriptografi ElGamal, yaitu sistem kriptografi yang menggunakan algoritma ElGamal, definisi himpunan-himpunan plainteks, cipherteks dan kunci, serta proses enkripsi dan dekripsi, seperti diberikan pada gambar berikut ini.

Diberikan bilangan prima p dan sebuah elemen primitif $\alpha \in \mathbb{Z}_p^*$. Ditentukan

$\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ dan $a \in \{0,1,\ldots, p-2\}$. Didefinisikan

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta = \alpha^a \bmod p\}.$$

Nilai p, α, dan β dipublikasikan, dan nilai a dirahasiakan.

Untuk $K = (p, \alpha, a, \beta)$, plainteks $m \in \mathbb{Z}_p^*$ dan untuk suatu bilangan acak rahasia $k \in \{0,1,2,\ldots, p-2\}$, didefinisikan

$$e_K(m, k) = (\gamma, \delta)$$

dengan

$$\gamma = \alpha^k \bmod p.$$

dan

$$\delta = \beta^k . m \bmod p.$$

Untuk $\gamma, \delta \in \mathbb{Z}_p^*$, didefinisikan

$$d_K(\gamma, \delta) = \delta . (\gamma^a)^{-1} \bmod p.$$

Gambar 2.1. Sistem Kriptografi ElGamal.

2.2 Enkripsi

Proses pertama adalah pembentukan kunci yang terdiri dari kunci rahasia dan kunci publik. Pada proses ini dibutuhkan sebuah bilangan prima p yang digunakan untuk membentuk grup ℤP*, elemen primitif α dan sebarang a ϵ {0,1,..., p - 2}. Kunci publik algoritma ElGamal berupa pasangan 3 bilangan, yaitu (p, α, β), dengan :

β=α^a mod p

Sedangkan kunci rahasianya adalah bilangan a tersebut.

Berikut ini diberikan suatu algoritma yang dapat digunakan untuk melakukan pembentukan kunci.

Algoritma Pembentukan Kunci

Input : Bilangan prima aman p > 255 dan elemen primitif a ϵ ℤP*.

Output : Kunci publik (p, α, β) dan kunci rahasia a.

Langkah :

1. Pilih a ϵ {0,1,..., p - 2}.

2. Hitung β=α^a mod p
3. Publikasikan nilai p, α, dan β , serta rahasiakan nilai a.

Pihak yang membuat kunci publik dan kunci rahasia adalah penerima, sedangkan pihak pengirim hanya mengetahui kunci publik yang diberikan oleh penerima, dan kunci publik tersebut digunakan untuk mengenkripsi pesan. Jadi, kentungan menggunakan algoritma kriptografi kunci publik adalah tidak ada permasalahan pada distribusi kunci apabila jumlah pengirim sangat banyak serta tidak ada kepastian keamanan jalur yang digunakan.

2.3 ERD(Entity Relational Diagram) dari Sistem

Pada proses ini pesan dienkripsi menggunakan kunci publik (*p*, a, b) dan sebarang bilangan acak rahasia *k* ϵ {0, 1, ..., *p*-2}. Misalkan *m* adalah pesan yang akan dikirim. Selanjutnya, *m* diubah ke dalam blok-blok, sehingga diperoleh plainteks *m*1, *m2*, ..., *mn* dengan *m*i ϵ {1, 2, ..., p-1}, *i* =1, 2, ..., *n* Proses enkripsi pada algoritma ElGamal dilakukan dengan menghitung:

$$\gamma = \alpha^k \ mod \ p$$

dan

$$\delta = \beta^k . m \ mod \ p$$

dengan rahasia *k* ϵ {0, 1, ..., *p*-2} acak. Diperoleh cipherteks (γ, δ).

Bilangan acak *k* ditentukan oleh pihak pengirim dan harus dirahasiakan, jadi hanya pengirim saja yang mengetahuinya, tetapi nilai *k* hanya digunakan saat melakukan enkripsi saja dan tidak perlu disimpan. Berikut adalah algoritma enkripsi.

Algoritma Enkripsi

Input: Pesan yang akan dienkripsi dan kunci publik *(p, α, β).*

Output: Chiperteks (γi, δi), *i* = 1, 2, …, n.

Langkah:

1. Pesan dipotong-potong ke dalam bentuk blok-blok pesan dengan setiap blok adalah satu pesan.
2. Konversikan masing-masing karakter ke dalam kode ASCII, maka diperoleh plainteks sebanyak *n* bilangan, yaitu $m_1, m_2, \ldots, m_n$.
3. Untuk *i* dari 1 sampai *n* kerjakan:
   3.1. Pilih sebarang bilangan acak rahasia *ki* ϵ {0,1, ..., *p*- 2}.
   3.2. Hitung $\gamma = \alpha^k \bmod p$
   3.3. Hitung $\delta = \beta^k . m \bmod p$.

Diperoleh cipherteks yaitu (γ, δ), *i* = 1, 2, 3, .., n. Salah satu kelebihan algoritma ElGamal adalah bahwa suatu plainteks yang sama akan dienkripsi menjadi cipherteks yang berbeda-beda. Hal ini dikarenakan pemilihan bilangan *k* yang acak. Akan tetapi, walaupun cipherteks yang diperoleh berbeda-beda, tetapi pada proses dekripsi akan diperoleh plainteks yang sama.

2.4 Dekripsi

Setelah menerima cipherteks (γ,), proses selanjutnya adalah mendekripsi cipherteks menggunakan kunci publik *p* dan kunci rahasia *a*. Dapat ditunjukkan bahwa plainteks *m* dapat diperoleh dari cipherteks menggunakan kunci rahasia *a*.

Diberikan (p, α, β) sebagai kunci publik dan a sebagai kunci rahasia pada algoritma ElGamal. Jika diberikan cipherteks (γ,), maka

$$m = \delta . (\gamma^a)^{-1} mod \ p$$

dengan *m* adalah plainteks.

Karena ℤP* merupakan grup siklik yang mempunyai order *p*-1 dan *a* ϵ {0, 1, ..., *p* - 2}, maka $(\gamma^a)^{-1} = \gamma^{-a} = \gamma^{p-1-a} \ mod \ p$.

Algoritma Deskripsi

Input: Chiperteks (γi,), *i* = 1, 2, …, n, kunci publik p dan kunci rahasia a.

Output: Pesan asli.

Langkah:

1. Untuk *i* dari 1 sampai *n* kerjakan:

1.1. Hitung $\gamma^{p-1-a} \bmod p$
1.2. Hitung $m_i = \delta.(\gamma^a)^{-1} \bmod p$

2. Diperoleh plainteks $m_1, m_2, ..., m_n$.
3. Konversikan masing-masing bilangan $m_1, m_2, ..., m_n$ kemudian hasilnya digabungkan kembali.

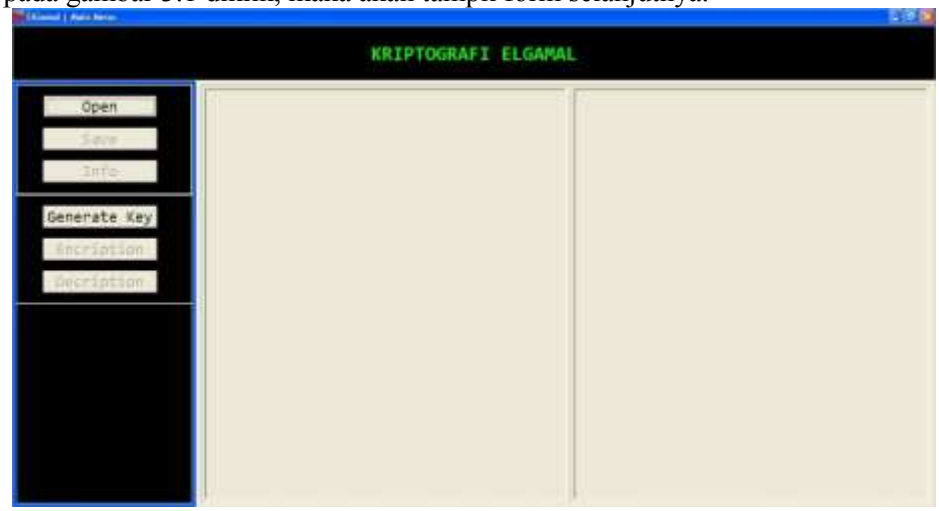
3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Hasil perncangan aplikasi adalah penjelasan hasil desain aplikasi yang sudah diterapkan di Borland Delphi. Berikut adalah hasil perancangan aplikasi.

Gambar 3.1 form pembuka

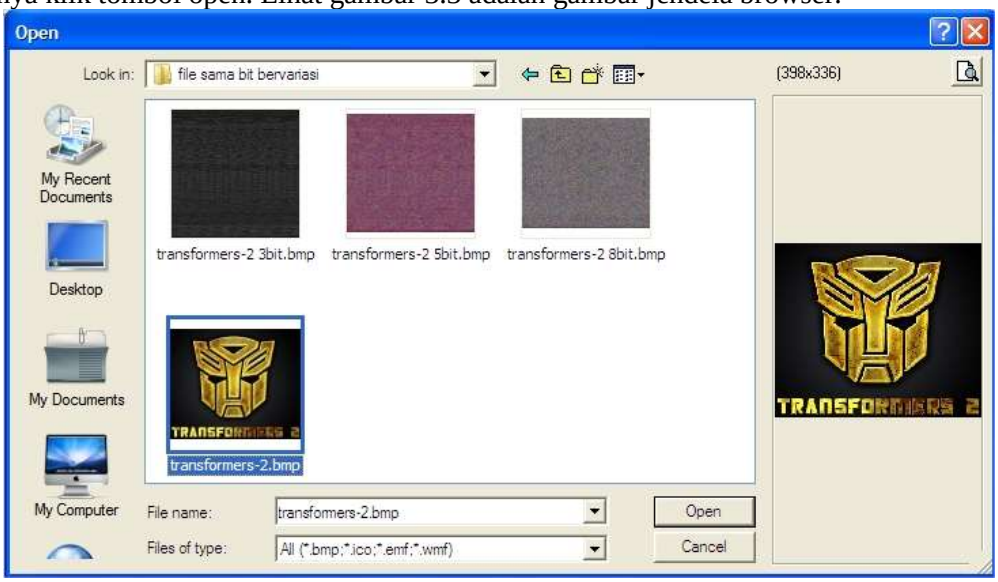
Gambar 3.1 adalah form pembuka ketika user menjalankan aplikasi. Form akan tampil di monitor jika icon atau menu desktop aplikasi diklik. Pertama kali, akan tampil dialog untuk melanjutkan penggunaan aplikasi lebih lanjut yaitu menuju form utama. Gambar 3.2 adalah tampilan form utama dimana aplikasi pemrosesan citra input dilakukan. Jika tombol 'Kriptografi' pada gambar 3.1 diklik, maka akan tampil form selanjutnya.



Gambar 3.2 tampilan form utama.

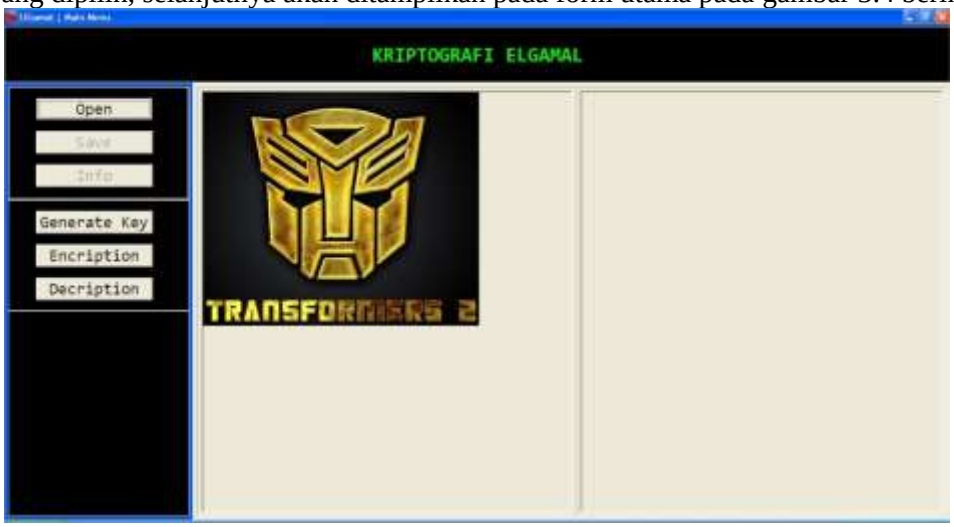
Pada form utama di gambar 3.2 pemrosesan dapat dikerjakan disini sesuai dengan menu-menu tombol yang disediakan. Beberapa pemrosesan adalah proses menginputkan citra, proses pembangkitan kunci, enkrpisi dan dekripsi.

Proses menginputkan citra asli dapat dilakukan pada menu yang disediakan yaitu dengan mengklik tombol open. Selanjutnya, akan muncul jendela browser untuk memilih satu citra input pada komputer. Citra input yang dipilih, akan diklik dan selanjutnya klik tombol open. Lihat gambar 3.3 adalah gambar jendela browser.



Gambar 3.3 Tampilan jendela browser.

Citra input yang dipilih, selanjutnya akan ditampilkan pada form utama pada gambar 3.4 berikut :

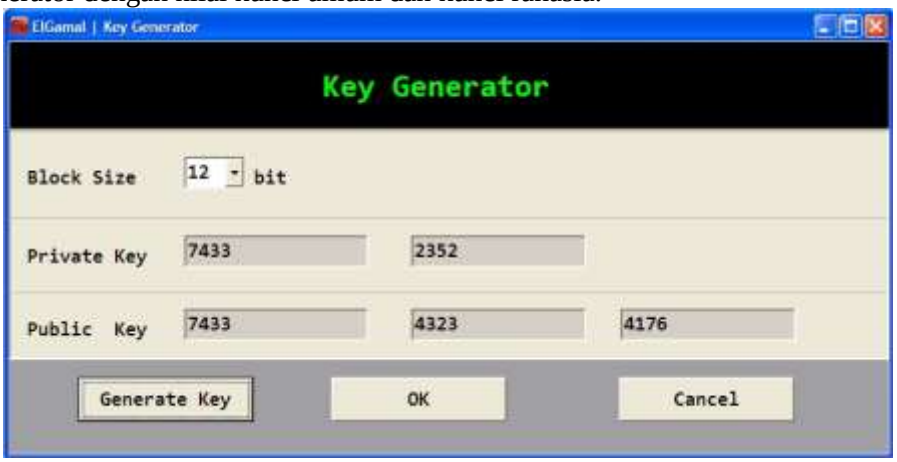


Gambar 3.4 tampilan citra input pada form utama.

Citra input selanjutnya dapat diproses dengan tombol-tombol pemrosesan yang siap dijalankan, tapi untuk memproses citra dibutuhkan kunci umum dan kunci rahasia. Pengguna bias menciptakan kunci dengan mengklik tombol Generate Key. Gambar 3.5 berikut adalah tampilan form Key Generator.

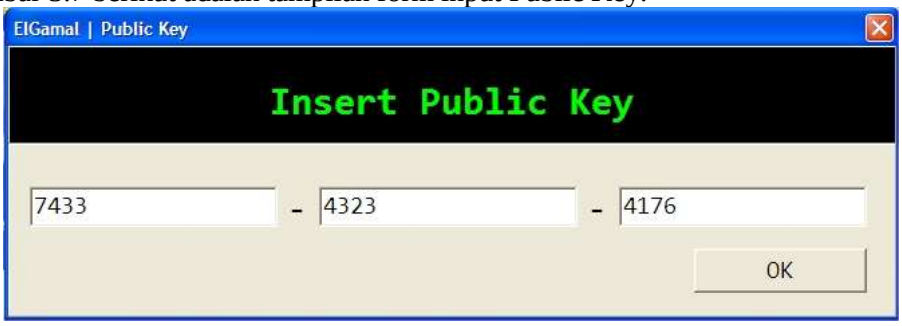
Gambar 3.5 Tampilan form Key Genereator

Pengguna dapat memilih nilai blocksize melalui Combobox yang ada di form tersebut. Pengguna bias menciptakan kunci sesuai dengan blocksize dengan cara mengklik tombol Generate Key. Gambar 3.6 berikut adalah tampilan form Key Generator dengan nilai kunci umum dan kunci rahasia.



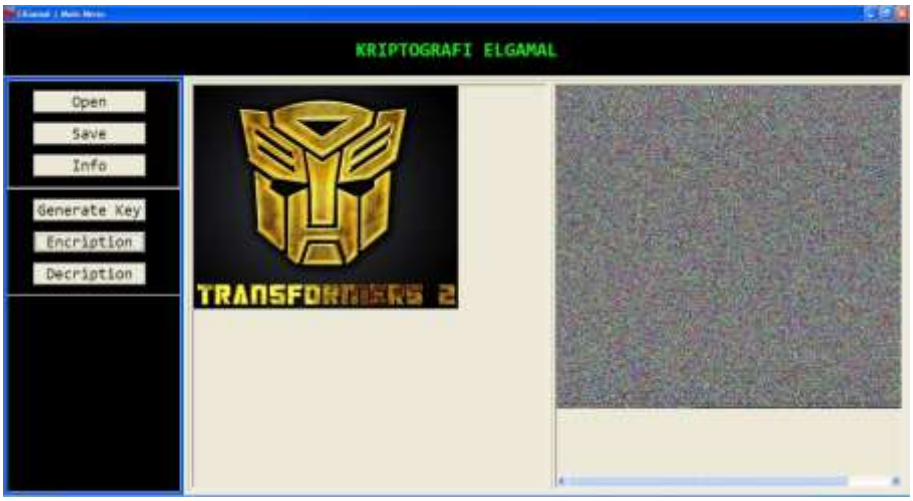
Gambar 3.6 Tampilan form Key Genereator dengan nilai kunci

Untuk proses enkripsi sendiri, pengguna harus memasukkan nilai kunci umum kedalam form Public Key yang sudah disediakan. Gambar 3.7 berikut adalah tampilan form input Public Key.



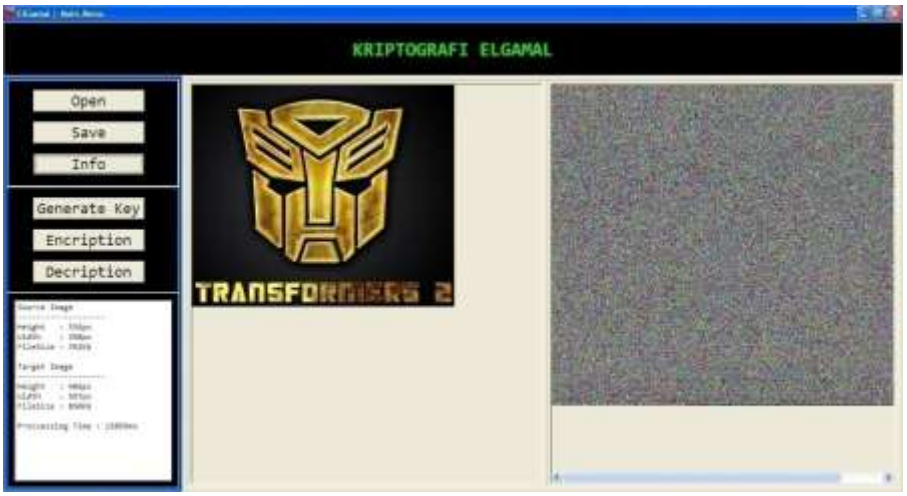
Gambar 3.7 Tampilan form input Public Key.

Citra input selanjutnya akan diproses. Hasil pemrosesan (citra ouput) juga ditampilkan pada form utama. Gambar 3.8 berikut adalah tampilan citra ouput pada form utama.



Gambar 3.8 Tampilan citra output pada form utama

Pada menu utama, terdapat tombol info yang difungsikan untuk menampilkan informasi meliputi informasi mengenai pemrosesan yaitu lokasi file, efektivitas dan waktu pemrosesan, Informasi dari citra sendiri terdapat pada form utama, dana hanya bias dilihat permrosesan selesai dan setelah tombol Info diklik. Tampilan info pada menu utama adalah sebagai berikut 3.9.

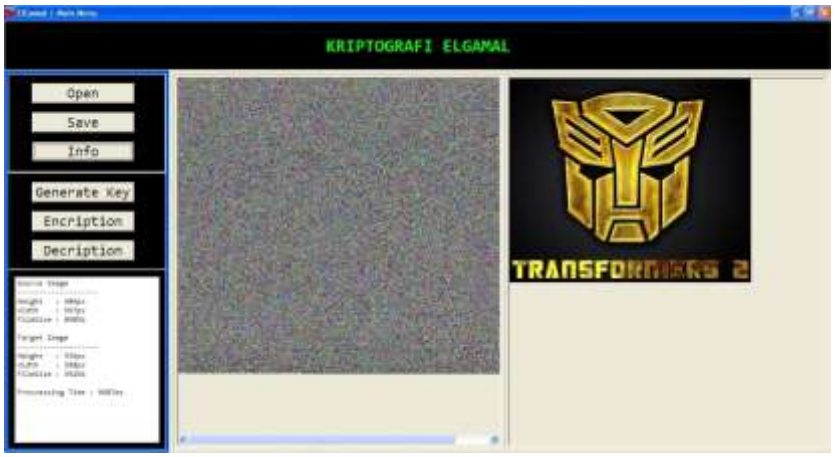


Gambar 3.9 Info pemrosesan.

Beberapa penjelasan diatas merupakan proses enkripsi, sedangkan proses dekripsi memiliki beberapa persamaan dengan proses enkripsi seperti pada menginputkan citra dan menampilkan info, namun proses dekripsi memiliki perbedaan pada input kunci, karna proses dekripsi membutuhkan kunci rahasia. Gambar 3.10 berikut adalah tampilan form input Private Key.

Gambar 3.10 Tampilan form input Private Key.

Berikut adalah contoh dari hasil proses dekripsi pada citra. Gambar 3.11 berikut adalah tampilan citra hasil dekripsi pada form utama.



Gambar 3.11 Tampilan citra hasil dekripsi pada form utama

3.2 Hasil Penelitian

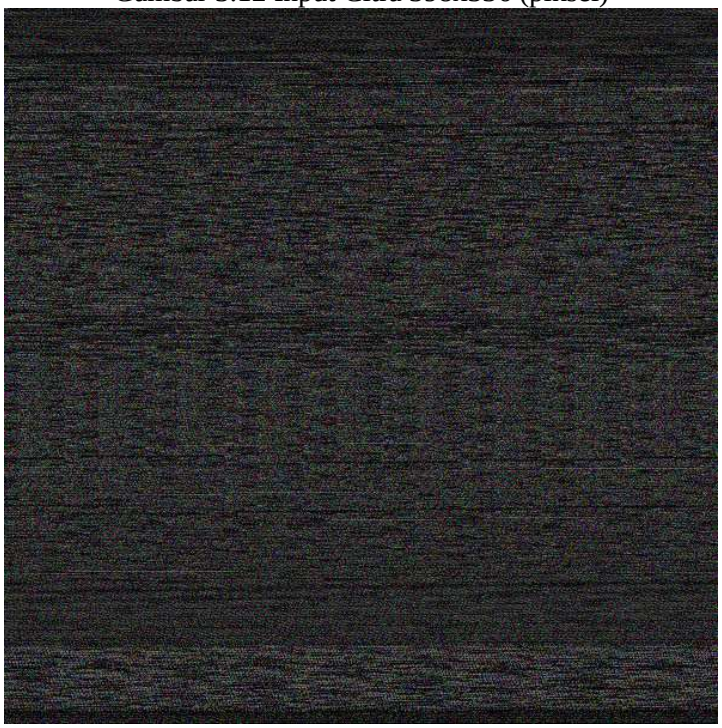
a. Enkripsi

Enkripsi pada citra dapat dilihat pada citra output. Algoritma *ElGamal* yang diterapkan pada aplikasi dalam memproses citra. Setiap kriteria akan diujikan pada aplikasi dengan citra input dengan ukuran atau resolusi 398x336 (piksel). Citra input yang diujikan tersebut berupa citra yang diujikan kepada nilai blocksize yang berbeda dengan penelitian sebagai berikut :

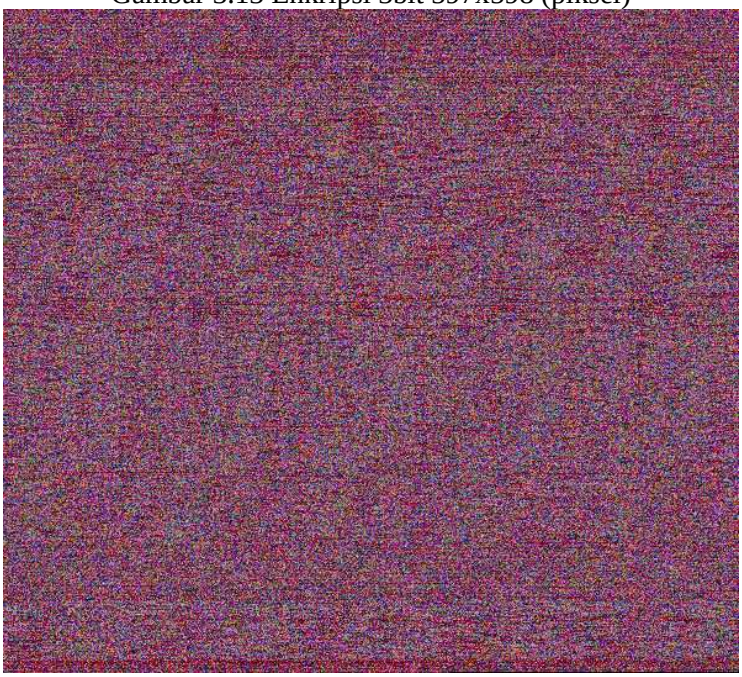
1. Blocksize berukuran 3bit, dengan kunci public (13,1,1)
2. Blocksize berukuran 5bit, dengan kunci public (47,41,3)
3. Blocksize berukuran 8bit, dengan kunci public (353,25,165)
4. Blocksize berukuran 12bit, dengan kunci public (7433, 4323,4176)

Hasil penelitian citra dengan kriteria BlockSize berukuran 3bit (gambar 3.13), BlockSize berukuran 5bit (gambar 3.14), BlockSize berukuran 8bit (Gambar 3.15) dan BlockSize berukuran 12bit (Gambar 3.16) adalah algoritma ElGamal dimana setiap plain atau data asli yang dimasukkan akan mencadi 2 buat cipher atau data yang tersandi.

Gambar 3.12 Input Citra 398x336 (piksel)



Gambar 3.13 Enkripsi 3bit 597x598 (piksel)



Gambar 3.14 Enkripsi 5bit 597x538 (piksel)

Gambar 3.15 Enkripsi 8bit 597x505 (piksel)

Gambar 3.16 Enkripsi 12bit 597x486 (piksel)

Pada percobaan diatas dapat diihat dimana citra asli (Gambar 3.12) telah mengalami proses enkripsi dengan menggunakan 4 nilai blocksize yang berbeda. Dapat diihat pada citra hasil enkripsi dimana citra hasil enkripsi mengalami perubahan dari segi tampilan maupun resolusinya. Dapat diihat juga bahwa antara nilai blocksize dengan resolusi citra hasil enkripsi memiliki nilai berbanding terbalik, dimana semakin kecil nilai blocksize maka resolusi citra akan semakin membesar.

Waktu pemrosesan

Pemrosesan citra input pada aplikasi akan membutuhkan lama waktu, sehingga perlu dilakukan pengamatan terhadap waktu pemrosesan. Beberapa kategori lama waktu pemrosesan dengan beberapa pengaruh adalah sebagai berikut :

i. Pengaruh BlockSize

Citra input yang diujikan pada lama waktu dalam pemrosesan yang dipengaruhi oleh blocksize diuji terhadap 4 buah nilai block, seperti pada table 4.1 berikut :

Tabel 3.1 kriteria blocksize

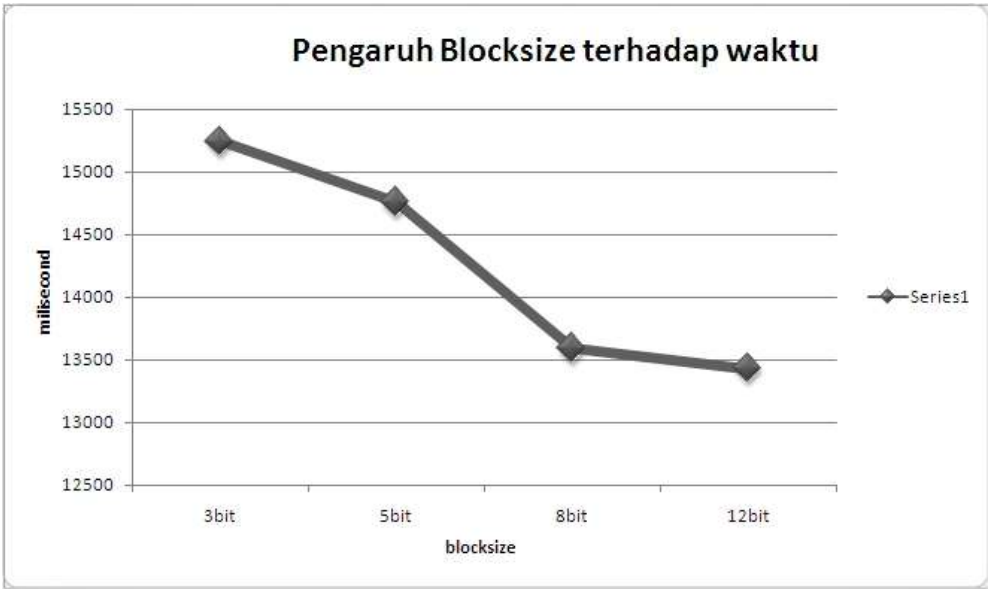
No	BlockSize	Kunci Publik
1	3bit	(13,1,1)
2	5bit	(47,41,3)
3	8bit	(353,25,165)
4	12bit	(7433,4323,4176)

Pada table 3.1 kriteria blocksize telah diujikan terhadap 4 biah nilai blocksize yang berbeda, untuk citra dan citra hasil enkripsi dapat dilihat pada Gambar 3.12, Gambar 3.13, Gambar 3.14, Gambar 3.15, Gambar 3.16. Berikut adalah pengujiannya:

Pengujian dilakukan sebanyak tiga kali, selanjutnya diperoleh nilai rata-rata. Hasil pengujian berupa lama waktu dijadikan table 4.2 dan Satuan yang digunakan adalah millisecond. Selanjutnya table tersebut dipresentasikan dalam grafik. Berikut adalah table 4.2 dan grafik 4.1 yaitu lama waktu pemrosesan dengan pengrauh nilai blocksize.

Tabel 3.2 waktu pemrosesan dipengaruhi oleh resolusi

	Waktu (milisec)			
Blocksize	1	2	3	rata-rata
3bit	14875	15000	15843	15239.33
5bit	14672	14890	14703	14755
8bit	13500	13844	13421	13588.33
12bit	13719	13078	13485	13427.33



Grafik 4.1 Waktu pemrosesan dipengaruhi oleh blocksize

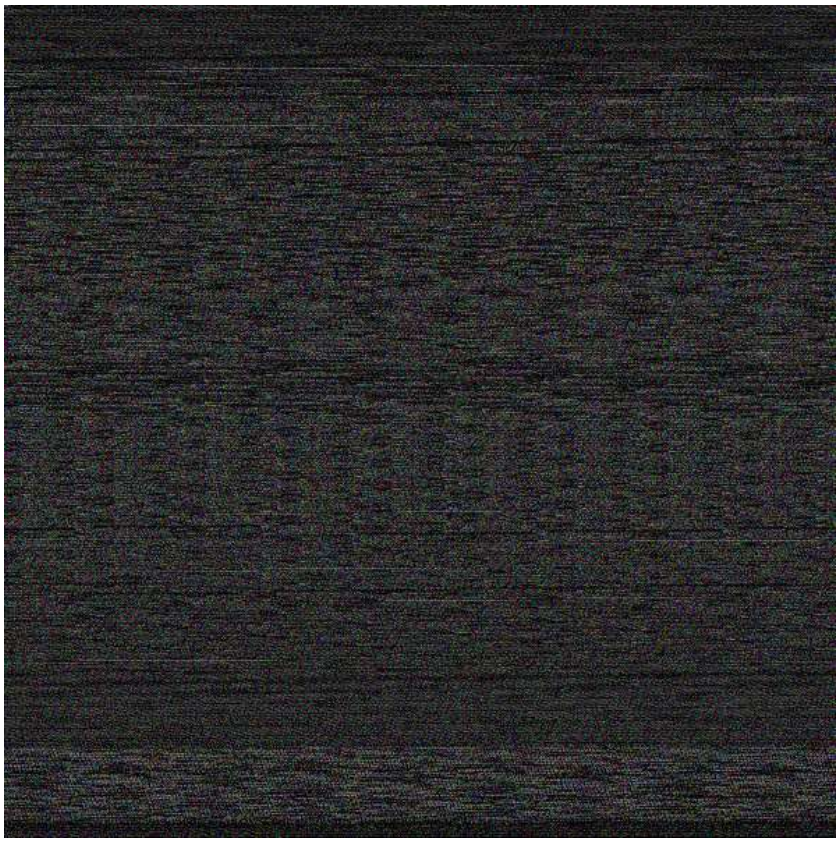
Pada hasil percobaan diatas dapat dilihat bahwa ada perbedaan waktu pemrosesan pada setiap nilai blocksize. Disini dapat dilihat juga bahwa nilai blocksize berbanding terbalik dengan waktu pemrosesan, dimana semakin kecil nilai blocksize maka semakin lama waktu yang diperlukan untuk memproses citra input.

b. Dekripsi

Dekripsi pada citra dapat dilihat pada citra output. Algoritma *ElGamal* yang diterapkan pada aplikasi dalam memproses citra. Setiap kriteria akan diujikan pada aplikasi dengan sumber citra input dengan ukuran atau resolusi 398x336 (piksel). Citra input yang diujikan tersebut berupa citra yang diujikan kepada nilai blocksize yang berbeda dengan penelitian sebagai berikut :

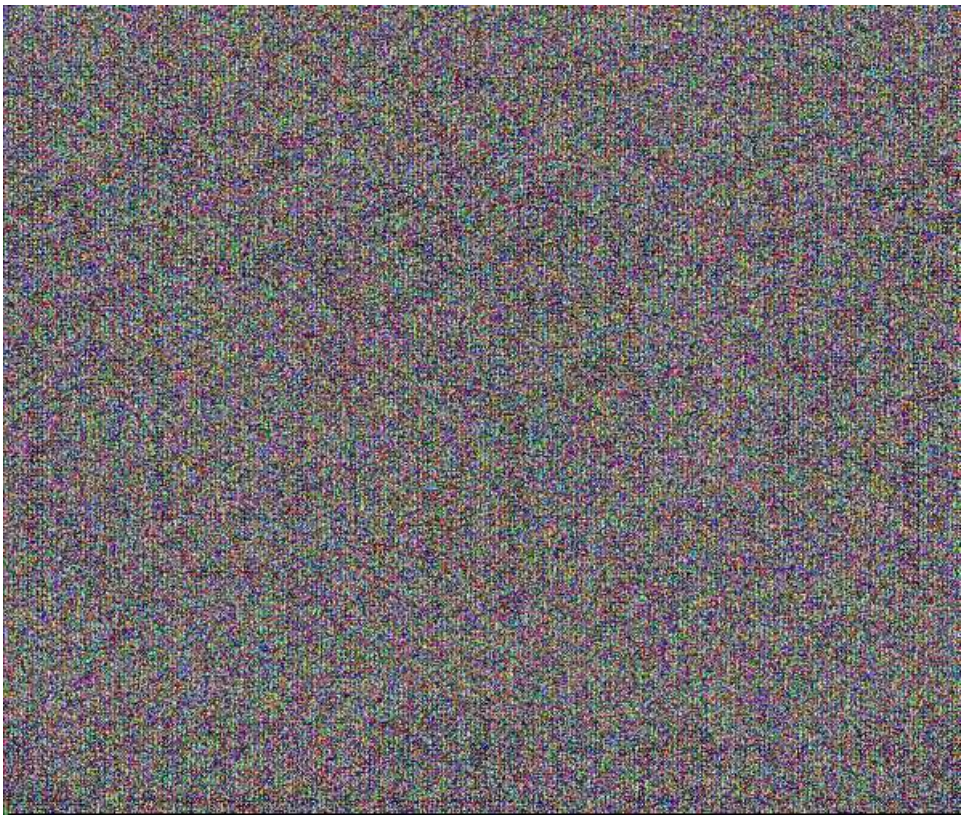
1. Blocksize berukuran 3bit, dengan kunci rahasia (13,3)
2. Blocksize berukuran 5bit, dengan kunci rahasia (47,32)
3. Blocksize berukuran 8bit, dengan kunci rahasia (353,53)
4. Blocksize berukuran 12bit, dengan kunci rahasia (7433,2352)

Citra Input merupakan hasil enkripsi dari citra yang sama namun dengan nilai blocksize yang berbeda. Penelitian yang dilakukan terhadap Gambar 3.23, Gambar 3.24, Gambar 3.25, Gambar 3.26 , adalah algoritma ElGamal dimana setiap 2 buah block cipher atau data tersandi yang dimasukkan akan mencadi sebuah plain atau data asli.



Gambar 3.23 Input Citra 3bit 597x598 (piksel)

Gambar 3.24 Input Citra 5bit 597x538 (piksel)



Gambar 3.25 Input Citra 8bit 597x505 (piksel)

Gambar 3.26 Input Citra 12bit 597x548 (piksel)

Gambar 3.27 Hasil Dekripsi dari masing Citra Input

Pada percobaan diatas dapat dilihat dimana 4 buah citra terenkripsi (Gambar 3.23 , Gambar 3.24, Gambar 3.25, Gambar 3.26) yang bersumber dari citra yang sama dan telah mengalami proses enkripsi dengan menggunakan 4 nilai blocksize yang berbeda. Dapat dilihat juga pada citra hasil dekripsi (Gambar 3.27) dimana citra hasil dekripsi identik dengan citra aslinya (Gambar 3.12).

Waktu pemrosesan

Pemrosesan citra input pada aplikasi akan membutuhkan lama waktu, sehingga perlu dilakukan pengamatan terhadap waktu pemrosesan. Beberapa kategori lama waktu pemrosesan dengan beberapa pengaruh adalah sebagai berikut :

Pengaruh BlockSize

Citra input yang diujikan pada lama waktu dalam pemrosesan yang dipengaruhi oleh blocksize diuji terhadap 4 buah nilai block, dimana citra input terenkripsi berasal dari hasil enkripsi citra yang sama, namun dengan nilai blocksize yang berbeda, seperti pada table 4.5 berikut :

Tabel 3.5 kriteria blocksize

No	BlockSize	Kunci Rahasia
1	3bit	(13,3)
2	5bit	(47, 32)
3	8bit	(353, 53)
4	12bit	(7433, 2352)

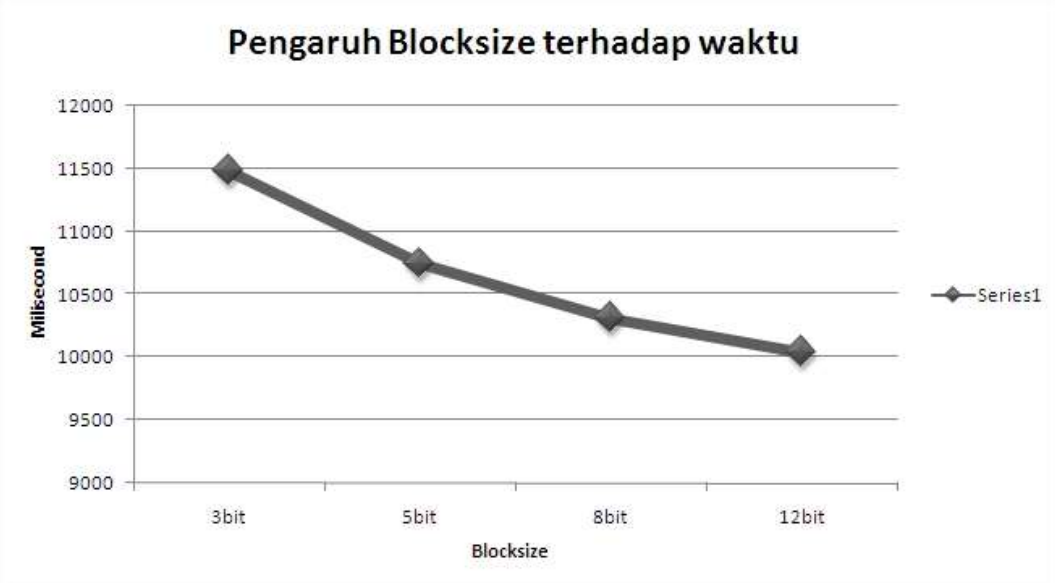
Pada table 3.5 kriteria blocksize telah diujikan terhadap 4 buah nilai blocksize yang berbeda, untuk citra input terenkripsi dan citra hasil dekripsi dapat dilihat pada Gambar 3.23, Gambar 3.24, Gambar 3.25, Gambar 3.26, Gambar 3.27. Berikut adalah pengujiannya:

Pengujian dilakukan sebanyak tiga kali, selanjutnya diperoleh nilai rata-rata. Hasil pengujian berupa lama waktu dijadikan table 4.6 dan Satuan yang digunakan adalah millisecond. Selanjutnya table tersebut

dipresentasikan dalam grafik. Berikut adalah table 4.6 dan grafik 4.3 yaitu lama waktu pemrosesan dengan pengrauh nilai blocksize.

Tabel 3.6 waktu pemrosesan dipengaruhi oleh resolusi

	Waktu (milisec)			
Blocksize	**1**	**2**	**3**	**rata-rata**
3bit	11671	11672	11110	11484.33
5bit	10609	10750	10875	10744.67
8bit	10188	10265	10485	10312.67
12bit	10110	9593	10422	10041.67



Grafik 4.3 Waktu pemrosesan dipengaruhi oleh blocksize

Pada hasil percobaan diatas dapat dilihat juga bahwa ada perbedaan waktu pemrosesan pada setiap nilai blocksize. Disini dapat dilihat juga bahwa nilai blocksize berbanding terbalik dengan waktu pemrosesan, dimana semakin kecil nilai blocksize maka semakin lama waktu yang diperlukan untuk memproses citra input.

c. Kriptoanalisis

Konversi Biner ke ASCII

Pada konversi biner ke ASCII akan dilakukan kriptoanalisis sederhana. Kriptoanalisis akan dilakukan pada gambar hasil enkripsi (Gambar 3.16). Langkah awal yang dilakukan adalah mengubah citra ke bilangan biner, lalu mengambil nilai setiap 8bit dan di konversikan ke ascii.

Tabel 3.9 Tabel Konversi Biner

Biner	Decimal	ASCII
10100010	162	ó
00110010	50	2
11100010	226	Γ
10000010	130	é
10110010	178	▓
10000010	130	é
00110010	50	2
01110001	113	q
10000000	128	Ç
00001010	10	◙

Pada Tabel 3.9 dapat dilihat bahwa pada 80 bit pertama tidak ada informasi yang dapat diambil dari hasil konversi ke kode ASCII.

Pembangkitan Kunci

Pada pembangkitan kunci, Algoritma *ElGamal* menggunakan logaritma diskrit, dimana bilangan a dipangkatkan dengan bilangan b lalu dimodulo dengan bilangan c. Sehingga perhitungan ini membutuhkan komputasi yang besar.

Pada Tabel 3.10 akan dilakukan pengujian terhadap kombinasi kunci dari blocksize terkecil sampai terbesar, namun perhitungan menggunakan perhitungan perpangkatan biasa tipe data yang digunakan adalah integer, dimana pada Delphi 7.0 tipe data integer memiliki jangkauan -2147483648…2147483647.

Tabel 3.10 Pembangkit Kunci

BlockSize	Prima	Publik 1	Privat	Publik 2
3bit	11	8	4	4
4bit	29	24	5	7
7bit	233	185	14	-100
9bit	701	223	26	-551
10bit	1879	1083	220	-225
13bit	13931	2467	11460	-10831

3.2 Pembahasan

Algoritma kriptografi *ElGamal* merupakan algoritma kriptografi block cipher, dimana dari data sumber diproses perblok-blok sesuai nilai blocksize yang ditentukan. Proses enkripsi nilai block selalu blocksize + 1 dan pada proses dekripsi nilai blocksize tetap.

Pada algoritma kriptografi *ElGamal*, sebuah data asli (plain) akan menjadi 2 buah data yang tersandi (cipher), itu sebabnya hasil enkripsi dari beberapa contoh diatas selalu hasil dari enkripsi akan menghasilkan citra terenkripsi yang 2 kali lebih besar dari citra aslinya ditambah jumlah dari dua kali blok cipher (dalam satuan bit). Pada proses enkripsi dan dekripsi waktu yang diperlukan untuk memproses sebuah citra sangat bergantung pada nilai blocksize yang dipilih. Ini dikarenakan semakin kecil nilai blocksizenya maka akan semakin banyak perulangan perhitungan algoritma ElGamal dan semakin besar pula citra hasil enkripsinya. Sedangkan papda proses dekripsi, gambar akan kembali lagi ke citra aslinya baik dari segi resolusi maupun ukuran file dan tampilannya. Tidak akan ada pixel yang terbuang atau terganti dengan nilai pixel yang berbeda

Hingga saat ini belum ada yang berhasil memecahkan algoritma ElGamal. Karena kekomplesitasan algoritma ini, maka penyerangan yang dilakukan dari segala sisi tidak mampu menembus pertahanan algoritma ElGamal ini.

Kelebihan dari algoritma ini adalah pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi dekripsi yang menggunakan proses komputasi yang besar sehingga hasil enkripsinya berukuran dua kali dari ukuran semula. Tetapi kekurangan dari algoritma ini adalah membutuhkan resource yang baik dan processor yang mampu untuk melakukan komputasi yang besar.

4. KESIMPULAN

Kesimpulan dari penelitian terhadap algoritma ElGamal dalam pemrosesan enkripsi dan dekripsi pada citra input. Berikut adalah kesimpulannya:

1. Algoritma kriptografi ElGamal dapat diimplementasikan terhadap file citra, karna hasil enkripsi citra input sangat berbeda dari citra inputnya, baik dari segi tampilan dan ukuran/resolusi.
2. Hasil enkripsi citra input selalu memiliki ukuran minimal 2 kali lebih besar dari citra aslinya.
3. Waktu pemrosesan enkripsi dan dekripsi dipengaruhi oleh blocksize dan resolusi, semakin kecil nilai blocksize dan semakin besar resolusi citra yang diproses maka waktu proses juga semakin lama.
4. Citra terenkripsi dapat kembali ke citra aslinya setelah mengalami proses dekripsi.
5. Proses dekripsi cenderung lebih cepat dari pada proses enkripsi.

5. SARAN

Untuk pengembanpgan, penelitian ini dapat dikembangkan dengan algoritma-algoritma kriptografi yang ada. Penelitian ini juga dapat dikembangkan kepada tipe berkas yang lain, seperti berkas bertipe suara dan berkas bertipe video.

DAFTAR PUSTAKA

[1] Ahmad, Usman, “Pengolahan Citra Digital, Graha Ilmu”, Yogyakarta, 2005.

[2] Achmad, Balza, Ir, M.Sc.E. dan Kartika Firdausy. ST, M.T. “Teknik Pengolahan Citra Digital Menggunakan Delphi”, Ardi Publishing, Yogyakarta, 2005.

[3] Antony, Pranata, “Pemrograman Borland Delphi” Andi Offset, Yogyakarta, 2001.

[4] Aniati Murni,Prof.Dr, dan Suryana Setiawan, “Pengantar Pengolahan Citra, Elex Media Komputindo, Kelompok Gramedia, Jakarta.

[5] zaki.math.web.id/download/skripsi/Skripsi-Algoritma_Kriptografi_ElGamal-Zaki_UGM.pdf , Tanggal Akses 10 Desember 2010

[6] Lucas J van Vliet, “Binary Image Processing”, http://www.ph.tn.tudelft.nl/~lucas