e-ISSN:2714-6529

Implementation of CAST-128 and RSA Algorithms on Tiered File Security Systems

[1]Fatmasari, [2]Rizky Tahara Shita, [3]Lauw Li Hin, [4]Marini
[1]Program Studi Sistem Informasi, STMIK Antar Bangsa
[2,3,4]Program Studi Teknik Informatika, Universitas Budi Luhur
Email: fsarie@gmail.com

Abstract

Data security in sending files needs to be a priority, especially for files sent that are confidential files; so we need a way to secure the data in the form of the file. Various methods and algorithms that can be used include the CAST-128 and RSA algorithms; where in the CAST-128 algorithm is a cryptographic algorithm using 16 feistel rounds which is one of its superior features which also in CAST-128 uses blocks of 64 bits long and a usable key length of 128 bits. Meanwhile, RSA is an easier algorithm to implement which can be used to support the CAST-128 algorithm in strengthening cryptographic algorithms to secure data in files. So that with the combination of the CAST-128 and RSA cryptographic algorithms, it is hoped that it can help improve data security in files that can be in the form of image, sound, video or file formats in other formats to avoid theft and theft of files sent via computer networks.

Keywords: *Kriptografi, Cast-128, RSA, Hybrid*

1. INTRODUCTION

Pengiriman data dalam file berformat melalui jaringan komputer adalah hal yang umum pada saat ini, baik file dalam format gambar, suara, video maupun format lainnya yang diperlukan bagi pribadi orang tersebut maupun secara perusahaan. Ada hal yang sering terlewatkan oleh pengguna biasa pada saat melakukan pengiriman file melalui jaringan komputer ini, yaitu keamanan terhadap file yang dikirimkannya. Sehingga selain edukasi yang baik kepada pengguna biasa, diperlukan juga sebuah mekanisme agar file yang dikirimkan tersebut dapat ditingkatkan dari sisi keamanannya pada saat dikirimkan melalui jaringan komputer dan internet.

Ilmu yang mempelajari agar file dapat diamankan dikenal dengan istilah kriptografi; yang mana memiliki tugas utamanya adalah menjaga kemananan terhadap file yang dilakukan kriptografi terhindar dari serangan orang yang tidak berkepentingan (penyadap/hacker). Ada banyak algoritma pada ilmu kriptografi yang dapat membantu meningkatkan keamanan file ini; antara lain adalah CAST-128 dan RSA. Kedua algoritma ini dapat digunakan dalam mengembangkan dan menerapkan implementasi tingkat keamanan file dalam bentuk aplikasi yang memudahkan pengguna, sehingga data dalam bentuk file ini lebih terjaga.

Berdasarkan jesisnya, terdapat 3 tipe algoritma kriptografi; Algoritma Simetri (Symetric Algorithms) disebut juga dengan algoritma kunci tunggal atau kunci rahasia (private key), Algoritma Asimetri (Asymetric Algorithms) yang dikenal dengan algoritma kunci publik (public key), Algoritma Hybrid yang merupakan gabungan dari Algoritma Simetri dan Asimetri.

1.1. Masalah

Permasalahan yang ada pada pengiriman file melalui jaringan komputer dan internet antara lain adalah cukup banyak celah yang dapat disusupi oleh hacker pada jalur komunikasi data, sehingga dapat berakibat pada bobolnya data dan informasi dan berkurangnya tingkat keamanan. Oleh karena itu, perlu dilakukan sebuah cara agar data yang dikirimkan dapat lebih aman dari sisi penerapan aplikasi.

1.2. Tujuan

Adapun tujuan dikembangkannya sebuah sistem keamanan file bertingkat dengan memanfaatkan algoritma CAST-128 dan RSA adalah untuk dapat mengamankan isi dari file atau data yang berupa text, gambar, audio, video maupun file dengan tipe lainnya agar dapat meminimalisir terhadap pencurian data maupun penyalahgunaan data yang dikirimkan melalui jaringan komputer maupun disimpan pada tempat penyimpanan pribadi (*storage*).

2. METHODS

2.1. Keamanan Data

Aspek penting dari sistem informasi adalah keamanan yang sering kali diabaikan dan dianggap sepele; salah satunya adalah tingkat keamanan selalu berbanding terbalik dengan kenyamanan dan performa. Hal ini dikarenakan memang dibutuhkan proses tambahan agar data dan file yang ingin diamankan dapat lebih baik dari sisi keamanannnya. Yang menjadi tantangan adalah bagaimana keamanan dapat ditingkatkan dengan tetap tidak terlalu mengganggu performa terhadap data maupun informasi yang sedang diproses. Dalam menjaga keamanan data dan informasi, maka perlu diketahui bahwa keamanan informasi adalah berfokus utama pada pencegahan penipuan yang dimulai dari melakukan deteksi terhadap sistem yang berbasis informasi [1].

2.2. Algoritma

Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi merupakan penulis buku arab yang oleh bangsa barat Al-Khuwarizmi dibaca Al-Gorism dan akhirnya berubah menjadi Algorithm yang didalam bahasa Indonesia menjadi Algoritma ini memiliki pengertian dalam urutan langkah yang secara logis dalam menyelesaikan masalah secara sistematis.

5 komponen utama dalam algoritma, yaitu: [2]

a) Finitness

   Algoritma yang harus berakhir setelah tidak berhingga.

b) Definiteness

   Merupakan langkah yang harus didefinisikan dengan tepat, tindakan yang dimuat harus teliti dan dengan jelas ditentukan untuk setiap kondisi.

c) Input

Merupakan masukan yang digunakan untuk diproses.

d) Output

Adalah hasil dari proses yang sudah dilakukan berdasarkan dari input yang dilakukan oleh pengguna.

e) Effectiveness

Agar dapat menjadi efektif untuk algoritma yang digunakan.

2.3. Kriptografi

Kriptos dan Graphia merupakan gabungan kata yang berasal dari Yunani dan menghasilkan kata kriptografi; dimana Kriptos digunakan untuk menjelaskan suatu yang disembunyikan atau rahasia, sedangkan Graphia berarti tulisan. Oleh karena itu, kriptografi merupakan ilmu yang mempelajari tulisan dapat dikomunikasikan secara rahasia dengan metode tertentu agar tidak mudah bocor oleh orang yang tidak berkepentingan dan hanya dapat dibaca oleh orang yang memiliki metode yang sama tersebut.

4 tujuan dasar ilmu kriptografi yang menjadi aspek keamanan informasi, antara lain: [3]

a) Confidentiality (Kerahasiaan)
b) Data Integrity (Integritas Data)
c) Authentication (Autentikasi)
d) Non-Repudiation (Non Penolakan)

2.4. Algoritma Hybrid

Dalam proses transfer data, metode simetris kurang tepat dikarenakan komunikasi data harus bertukar kunci yang dibuat secara acak untuk setiap session key; jadi jika terdapat seorang peretas yang menemukan kunci kriptografi, maka dapat dengan mudah melakukan proses dekrip. Untuk mengatasi permasalahan itu, maka metode asimetris digunakan dengan membuat pasangan kunci. Dengan adanya pasangan kunci ini, maka pengirim memprosesnya dengan menggunakan public key dan private key digunakan penerima pada proses dekripsi data, sehingga kunci untuk proses dekripsi tidak jatuh ke orang lain. Private key pada metode asimetris juga bukan merupakan turunan dari public key; yang meskipun hal tersebut telah dilakukan, metode asimetris juga mempunyai kekurangan yaitu memiliki proses seribu kali lebih lambat jika dibandingkan dengan metode simetris dan hal ini tidak cocok untuk memproses data dalam jumlah besar.

Algoritma hybrid dapat dijelaskan sebagai sebuah sistem yang melakukan penggabungan metode simetris dan asimetris. Dimulai dengan cipher asimetris, dimana kedua belah pihak setuju dengan private key (session key) yang akan digunakan dan kemudian digunakan metode simetris untuk

melakukan enkripsi pesan dan private key hanya dapat dipakai 1 sesi saja dan pada proses berikutnya harus dibuat private key kembali.

2.5. CAST-128

Salah satu algoritma pada bidang ilmu kriptografi adalah CAST yang dikembangkan oleh Carlisle Adams dan Standard Tavares. Pada algoritma CAST terdapat 2 macam; yaitu: CAST-128 & CAST-256. Pada algoritma CAST-128 menggunakan 12 atau 16 putaran (round) feistel dimana 64 bit ukuran blok dan kunci dikisaran 40 hingga 128 bit. Komponen yang ada didalam sebuah 8 x 32 bit Sbox yang berdasarkan dari bent-function, rutasi kunci independen, modular addition, substraksi dan operasi XOR dan pada CAST-128 ini termasuk dalam kelas algoritma enkripsi yang menggunakan feistel. [4]

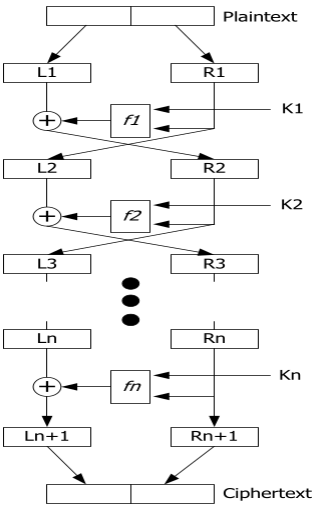

Figure 1: Putaran Feistel

Beberapa aplikasi yang menggunakan CAST-128, antara lain: [4]

a) Telnet
b) Cryptographic Message Syntax (CMS)
c) Encapsulating Security Payload (ESP)

2.6. RSA

Pada tahun 1977, RSA dikembangkan oleh Ron Riverst, Adi Shamir dan Leonard Adleman yang kemudian nama mereka digunakan sebagai singkatannya (Rivest Shamir Adleman – RSA) dan dipatenkan pada tahun 1983 oleh MIT dimana masa berlakunya paten tersebut adalah sampai 21 September 2000; sehingga RSA dapat digunakan secara bebas saat ini. Proses RSA yang mudah dimengerti, sehingga algoritma ini dapat dengan mudah diimplementasikan [5].

Proses pemfaktoran bilangan yang besar dan sulit pada algoritma RSA, membuatnya menjadi sebuah standar kriptografi asimetris yang mana menggunakan 2 key; yaitu private key dan public key.

Public key ini digunakan untuk melakukan proses enkripsi informasi sumber; dan private key digunakan pada proses dekripsi yang membuatnya menjadi dapat dibaca kembali. Pada RSA, proses enkripsi dan dekripsi dilakukan secara bolak – balik dan pada proses RSA dibagi menjadi 3 bagian; yaitu: membangkitkan kunci, algoritma enkripsi dan algoritma dekripsi.

2.6.1. Proses Enkripsi RSA

Proses enkripsi pada RSA dilakukan dengan menentukan public key (e) dan modulus (n) dari sebuah pesan yang kemudian akan memecah plain text menjadi blok (m1, m2, ...). Pada setiap blok m1 dilakukan proses enkripsi menjadi blok ci yang akan digabung dan diberikan sebagai kunci publik. Gambar berikut ini menjelaskan alur proses enkripsi RSA tersebut:

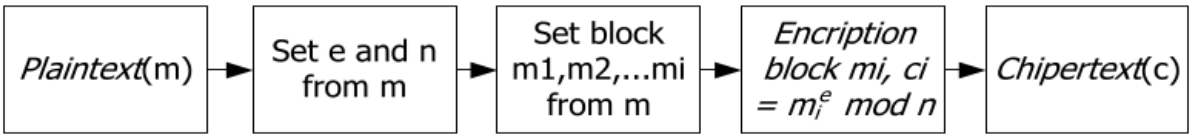

Figure 2: Proses Enkripsi RSA

2.6.2. Proses Dekripsi RSA

Untuk proses dekripsi RSA, maka pesan yang diterima dalam bentu terenkripsi (chipertext) dengan memproses setiap blok ciphertext (ci) dilakukan proses dekripsi menjadi blok mi. Proses dekripsi dilakukan dengan menggunakan kunci private d = 1019 yang kemudian dilakukan konversi kembali menggunakan karakter ASCII. Gambar berikut menjelaskan proses dekripsi RSA tersebut:

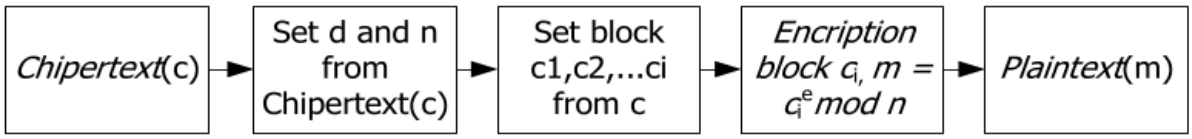

Figure 3: Proses Dekripsi RSA

2.7. Kelebihan dan kekurangan RSA

2.7.1. Kelebihan RSA

Adapun kelebihan dari algoritma RSA, antara lain adalah: [5]

a) Tingkat kesulitan pada Algorimta RSA adalah terdapat pada proses pemfaktoran bilangan menjadi faktor prima. Dalam hal ini, proses dilakukan dengan memfaktorkan r menjadi p dan q. Sebab saat r berhasil dilakukan pemfaktoran, maka melakukan perhitungan m akan lebih mudah.

b) Pada proses brute force, algoritma RSA memiliki ketahanan yang lebih baik. Ini disebabkan pada proses dekripsi yang kompleks dimana ditentukan secara dinamis dalam menentukan p dan q untuk proses pembangkitan kunci dan menghasilkan key space yang besar dan berdampak bagus terhadap serangan.

2.7.2. Kekurangan RSA

Sedangkan kekurangan dari algoritma RSA, yaitu: [5]

a) Dibandingkan dengan DES dan AES, RSA memiliki proses yang lebih lambat.
b) Melakukan proses enkripsi kunci simetri dengan kunci publik pada penerima pesan.
c) Proses enkripsi dilakukan dengan algoritma simetri.
d) Pesan dan private key dikirimkan secara bersamaan.
e) Pada proses dekripsi, penerima menggunakan private key dan melakukan prosesnya menggunakan kunci simetri.

3. RESULTS AND DISCUSSION

3.1. Strategi Pemecahan Masalah

Dengan adanya berbagai jenis algoritma kriptografi, penulis dalam hal ini melakukan beberapa strategi untuk mengamankan informasi dengan cara melakukan penggabungan dua jenis algoritma yang berbeda yaitu algoritma simetris dan asimetris. Dari berbagai macam jenisnya penulispun menganalisa dan membaca dari media seperti internet, buku dan lainnya untuk mengetahui algoritma manakah yang sedikit banyak dibicarakan dalam masalah kekuatan kunci untuk mengamankan sebuah informasi. Pada akhirnya penulis memilih algoritma CAST-128 sebagai algoritma simetris dan algoritma RSA sebagai algoritma asimetris sehingga menjadi algoritma jenis hibrid. Gabungan teknik pengacakan informasi atau data ini diharapkan dapat lebih memperkuat pengamanan akan sebuah informasi yang berupa file text, file image, file audio, file video dan lain-lain seperti file database.

3.2. Skema Proses Enkripsi

Pada ilmu kriptografi, sebuah proses penyandian data dan informasi dengan algoritma tertentu yang digunakan untuk meningkatkan keamanan; dimana pada proses enkripsi yang digunakan adalah dengan menggabungkan Algoritma CAST-128 dengan satu private key dan Algorimta RSA yang membutuhkan 2; yaitu public key dan private key. Proses enkripsi ini membutuhkan file sumber dan kata sandi yang didefinisikan oleh pengguna. Proses yang dihasilkan dari kedua Algoritma ini adalah file baru yang sudah terenkripsi dan sebagai key yang digunakan adalah private key dari Algoritma RSA. Skema proses enkripsi tersebut adalah sebagai berikut:

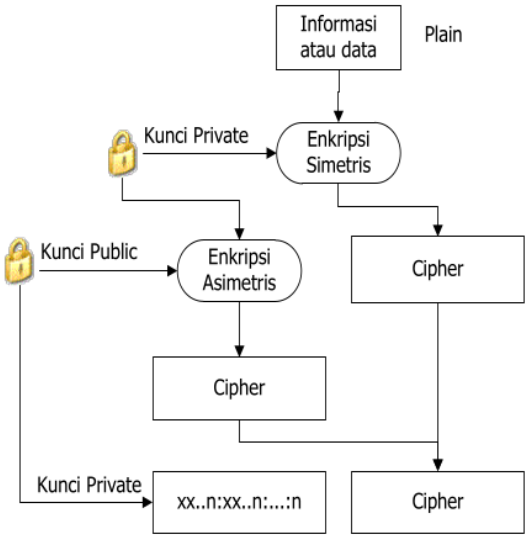

Figure 4: Skema Proses Enkripsi

3.3. Skema Proses Dekripsi

Pengembalian informasi dari data yang sudah dilakukan proses enkripsi dikenal dengan istilah dekripsi. Karena aplikasi menggunakan 2 algoritma (CAST-128 dan RSA), maka pertama kali akan dilakukan pemilihan file sebagai sumber yang akan dikenakan proses dekripsi tersebut yang berikutnya membutuhkan informasi dari private key agar file dapat dilakukan proses dekripsi dengan sesuai. Adapun hasil yang dilakukan pada tahap ini adalah file sumber yang sudah terenkripsi dapat menjadi dibaca secara normal kembali.

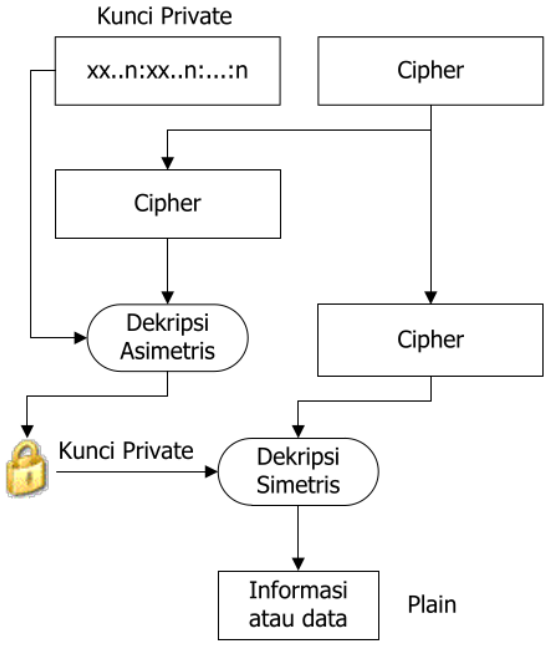

Figure 5: Skema Proses Dekripsi

3.4. Entity Relationship Diagram (ERD)

Basis data dibuat untuk menyimpan informasi atau data yang telah dienkripsi dan juga didekripsi beserta kuncinya. Sistem akan melakukan penyimpanan apabila pengguna atau user menginginkannya. Entity Relation Diagram merupakan diagram yang menjelaskan struktur data yang terdapat di dalam sistem dan menunjukkan relasi antar tabel.

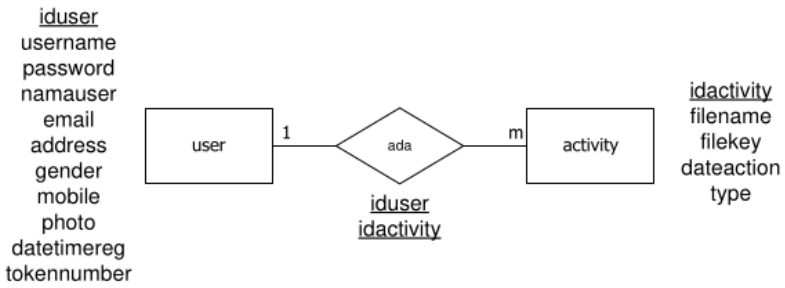

Figure 6: Entity Relationship Diagram

3.5. Proses Enkripsi

Diagram alir proses enkripsi ini menjelaskan urutan-urutan dimana suatu informasi atau data dienkripsi dari plain menjadi cipher dengan input-an dari user atau pengguna yaitu, panjang BIT berupa bilangan dan data yang akan dienkripsi dan juga terdapat proses untuk penghapusan data yang telah dienkripsi sebelumnya dalam database.

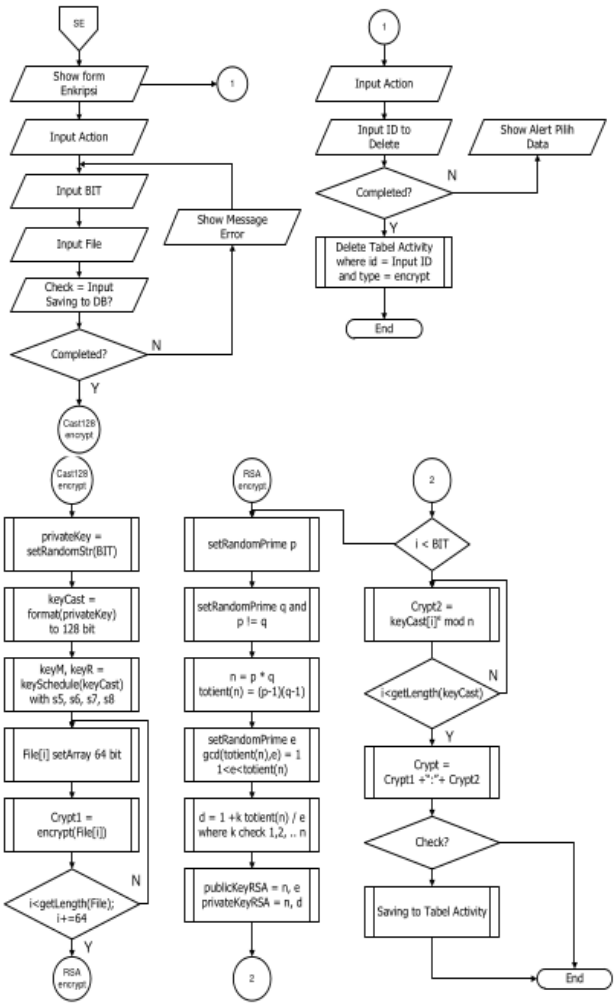

Figure 7: Flowchart Enkripsi

3.6. Proses Dekripsi

Diagram alir proses dekripsi ini menjelaskan urutan-urutan dimana suatu informasi atau data didekripsi dari cipher menjadi plain dengan input-an dari user atau pengguna yaitu, file kunci pembuka dan informasi atau data yang telah di enkripsi berupa file dan juga terdapat proses untuk penghapusan data yang telah didekripsi sebelumnya dalam database.

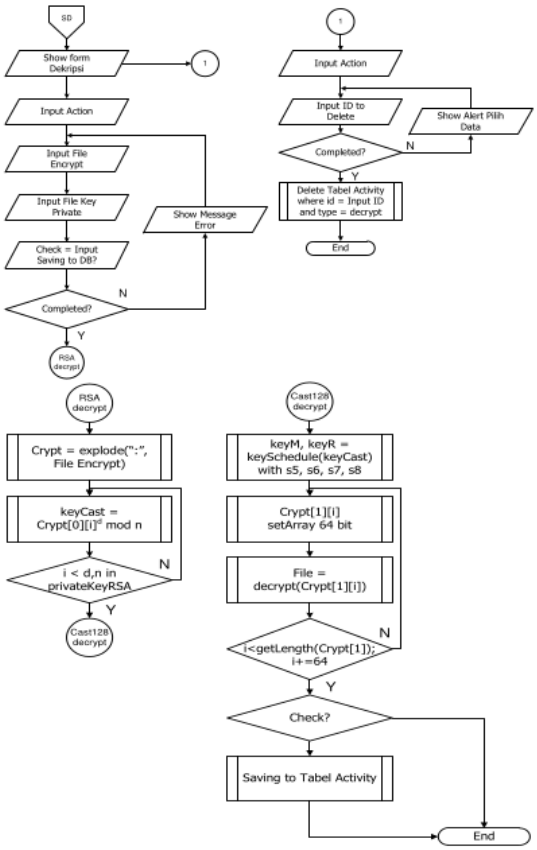

Figure 8: Flowchart Dekripsi

3.7. Aplikasi Kriptografi untuk Proses Enkripsi

Untuk proses pengenkripsian pengguna atau user cukup memasukkan atau meng-input BIT untuk dijadikan acuan kunci dan file yang akan di enkripsi setelah itu user diharuskan menglik tombol Enkripsi File untuk proses pengenkripsian. Ketika tombol telah diklik maka aplikasi akan menanyakan Simpan enkripsi anda ? Dimana jika user memilih ya, maka otomatis aplikasi menyimpan hasil enkripsi beserta kuncinya ke dalam database kemudian aplikasi akan menampilkan listing berbentuk tabel data yang telah disimpan, jika user memilih tidak maka aplikasi hanya menampilkan tombol untuk men-download hasil enkripsi beserta kuncinya.

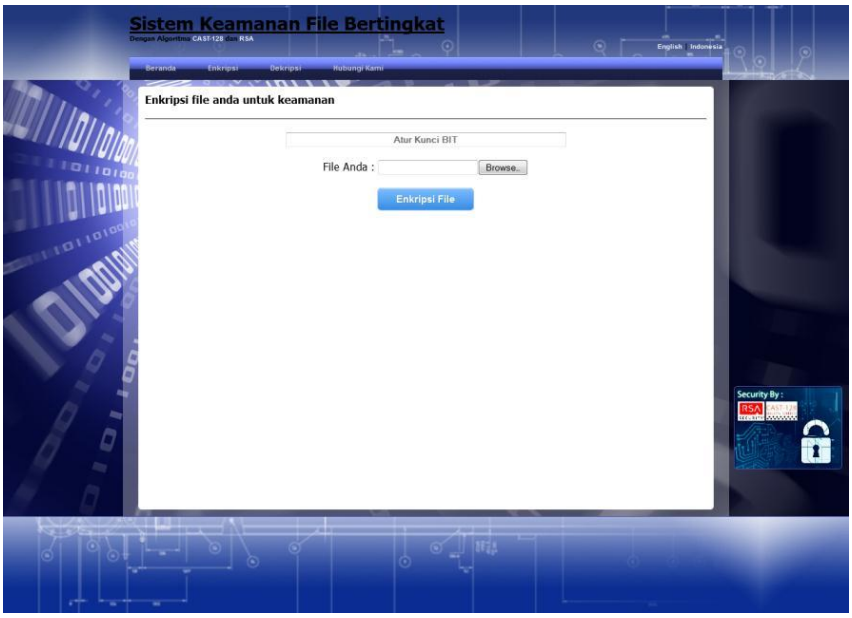

Figure 9: Tampilan aplikasi pada proses enkripsi

3.8. Aplikasi Kriptografi untuk Proses Dekripsi

Untuk proses pendekripsian sama halnya seperti proses pengenkripsian hanya saja input yang dimasukkan berbeda yaitu satu file yang telah dienkripsi sebelumnya dan satu lagi file kunci pembuka kemudian proses dekripsi akan diolah oleh aplikasi dengan mengklik tombol Dekripsi File.

Figure 10: Tampilan aplikasi pada proses dekripsi

3.9. Pengujian Aplikasi Kriptografi

Pengujian program dilakukan dengan berbagai jenis Input File seperti document, image, audio dan video; yaitu:

Table 1 Pengujian Proses Enkripsi

Input File	Output File	BIT	Selisih	Waktu (ms)
BAB III.doc 2,12 MB	BAB III.doc.cast128rsa 5,03 MB	5	2,91 MB	33,04
ElGamal.pdf 0,98 MB	ElGamal.pdf.cast128rsa 2,34 MB	10	1,36 MB	17,55
TA.ppt 526 KB	TA.ppt.cast128rsa 1,21 MB	15	684 KB	11,7
Img 1.jpg 1,96 MB	Img 1.jpg.cast128rsa 4,64 MB	5	2,68 MB	31,05
Nano.mp3 1,01 MB	Nano.mp3.cast128rsa 2,41 MB	5	2,78 MB	33,18
Billyard.3gp 987 KB	Billyard.3gp.cast128rsa 2,28 MB	15	1,293 MB	22,56

Sedangkan pada proses dekripsi, hasil pengujiannya adalah sebagai berikut:

Table 2 Pengujian Proses Dekripsi

Input File	Output File	Selisih	Waktu (ms)
BAB III.doc.cast128rsa 5,03 MB	BAB III.doc 2,12 MB	2,91 MB	33,22
ElGamal.pdf.cast128rsa 2,34 MB	ElGamal.pdf 0,98 MB	1,36 MB	16,28
TA.ppt.cast128rsa 1,21 MB	TA.ppt 526 KB	684 KB	10,74
Img 1.jpg.cast128rsa 4,64 MB	Img 1.jpg 1,96 MB	2,68 MB	31,6
Nano.mp3.cast128rsa 2,41 MB	Nano.mp3 1,01 MB	2,78 MB	18,62
Billyard.3gp.cast128rsa 2,28 MB	Billyard.3gp 987 KB	1,293 MB	24,72

3.10. Analisa Hasil Pengujian

Berdasarkan pengujian program untuk poses enkripsi dan dekripsi diatas, dapat dijelaskan beberapa hal sebagai berikut :

a) Algoritma CAST-128 dan Algoritma RSA dapat diterapkan pada aplikasi yang berjalan diatas browser.
b) Program aplikasi dapat mengenkripsi dan mendekripsi semua jenis tipe file yaitu document, image, audio dan video.
c) Ukuran file output menjadi lebih besar dari file input pada proses enkripsi, dan besarnya ukuran file tergantung dari panjang kunci bit yang di input dan jenis file yang disebabkan karakteristik kerapatan bit setiap file-nya berbeda.
d) File hasil dari dekripsi tidak mengalami kerusakan atau sama dengan file aslinya ketika sebelum di enkripsi.
e) Kecepatan proses enkripsi dan dekripsi tergantung dari spesifikasi software, hardware, koneksi internet, besar file dan panjang kunci yang digunakan oleh pengguna aplikasi.

4. CONCLUSION

Berdasarkan dari analisa dan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa pengamanan informasi dapat menggunakan algorithm bertipe hybrid, dimana dalam kasus ini merupakan penggabungan algoritma CAST-128 dan RSA dimana kecepatan pengolahan kriptografi dapat dinyatakan optimal yang juga bergantung pada hardware yang digunakan. File hasil proses kriptografi disimpan dalam database yang dibuat khusus, serta pemanfaatan kunci kriptografi yang semakin panjang akan berdampak pada lamanya proses kriptografi dan hasil file tersebut.

REFERENCES

[1] S. Garfinkel, G. Spafford, and A. Schwartz, *Practical UNIX and Internet Security*, no. April. California: O'Reilly & Associates, Inc., 2003. [Online]. Available: http://books.google.com.br/books?id=t0IExLP-MPMC

[2] D. E. Knuth, *The Art of Computer Programming*, Third. Boston: Pearson Education, Inc., 2014.

[3] E. R. Agustina and A. Kurniati, "Pemanfaatan Kriptografi dalam Mewujudkan Keamanan Informasi pada e-Voting Indonesia," *Semin. Nas. Inform.*, pp. 22–28, 2009.

[4] A. Gunawan, "Studi Mengenai Algoritma Simetri CAST-128 dan Aplikasinya," *Inst. Teknol. Bandung*, pp. 1–18, 2006.

[5] M. Iqbal, "Studi Teknis Metode Enkripsi RSA dalam Perhitungannya," *Inst. Teknol. Bandung*, 2006.