IMPLEMENTASI ALGORITMA BLOWFISH DAN ALGORITMA RC4 PADA APLIKASI KEAMANAN EMAIL

Hatanu Prasidya Martosedono[1], Purwanto[2]
[1,2)]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur.
Jl. Raya Ciledug, Petukangan Utara, Jakarta Selatan
Telp. (021) 5853753, Fax (021) 5866369
E-mail: hatanuprasidya@gmail.com[1], purwanto@budiluhur.ac.id[2]

ABSTRAKSI

Menjaga keamanan dan kerahasiaan data atau informasi dalam email merupakan sebuah aspek penting yang dipermasalahkan. Ada banyak ancaman dari pihak ketiga yang mengancam keamanan data atau informasi yang tersimpan di dalam email. Berdasarkan hal tersebut, maka perlu dibuat aplikasi kriptografi email menggunakan algoritma Blowfish dan RC4. Aplikasi yang akan dibuat ini akan mengenkripsikan file (plaintext) yang mempunyai ekstensi .txt, .doc, .docx, .xlsx, .xls dan .pdf dengan menambahkan password pilihan pada pesan dan file yang akan dienkripsi oleh aplikasi tersebut. Aplikasi ini akan mengenkripsi file dan pesan dalam email menjadi sebuah ciphertext (terenkripsi atau tersandi) agar data atau informasi yang ada tidak dapat dibaca atau dimengerti. Pesan dan file yang telah dienkripsi menjadi ciphertext bisa dikembalikan seperti semula jika didekripsi menggunakan password yang sama pada saat melakukan enkripsi file dan pesan tersebut. Metode penelitian yang digunakan untuk membangun aplikasi ini adalah waterfall. Hasil dari penelitian implementasi algoritma Blowfish dan RC4 pada aplikasi keamanan email ini telah terbukti fungsi dan kegunaannya untuk menjaga keamanan dan kerahasiaan informasi di dalam email dari pihak ketiga.

Kata kunci : Kriptografi , Keamanan Data, Enkripsi , Dekripsi, Blowfish, RC4 ,Waterfall.

1. PENDAHULUAN

1.1. Latar Belakang

Saat ini email tidak hanya dipakai sebagai sarana untuk berkomunikasi jarak jauh, namun kini juga banyak digunakan untuk membantu mengirim pesan dan informasi penting. Akibatnya, informasi yang disimpan di dalam email memerlukan pengamanan yang dapat melindungi keamanan informasi tersebut, untuk melindungi keamanan tersebut dapat dengan menggunakan enkripsi. Enkripsi adalah suatu metode pengamanan data/informasi dengan cara memasukkan data/informasi ke dalam suatu sandi dengan maksud menyembunyikan data/informasi tersebut. SMA Sumbangsih yang bergerak dalam bidang pendidikan memerlukan suatu cara pengamanan untuk menjaga kerahasiaan informasi yang penting seperti data sekolah dan pendidikan yang tersimpan di dalam email. Dengan adanya pertukaran informasi yang berisi soal ujian, data sekolah dan hasil pendidikan para murid maka dibutuhkan pengamanan email dengan attachment, dengan tujuan untuk mengelola dan mengamankan informasi konten email yang menyangkut tentang data sekolah. Hal inilah yang membuat penulis tertarik untuk merancang aplikasi keamanan email dengan attachment dokumen pada *email*. Pada aplikasi keamanan *email*, menggunakan algoritma *Blowfish* dan *RC4* tersebut walau tetap menjaga fungsi utama dari aplikasi tersebut.

1.2. Permasalahan

Berdasarkan uraian yang telah dijelaskan pada sub bab sebelumnya penelitian ini adalah bagaimana caranya menciptakan aplikasi keamanan email yang sederhana dan bagaimana caranya mengimplementasikan algoritma *Blowfish* dan algoritma *RC4* pada fungsi kriptografi di dalam aplikasi.

1.3. Batasan Masalah

Untuk menjaga lingkup dari penelitian ini berada dalam topik yang diajukan, maka beberapa batasan perlu didefinisikan antara lain:

a. Aplikasi ini akan dikembangkan sebagai aplikasi berbasis desktop, dengan menggunakan *Blowfish* sebagai algoritma utama dalam proses pengembangannya dan ditulis dalam bahasa pemrograman Java.
b. Aplikasi ini juga akan menggunakan *RC4* untuk algoritma kedua dalam pengembangan, tetapi implementasi algoritma utama akan diusahakan untuk tetap terpakai dalam pengembangan aplikasi ini.

c. Enkripsi dan dekripsi yang menjadi fungsi dalam aplikasi ini akan menggunakan algoritma Blowfish sebagai algoritma utama dan algoritma *RC4* sebagai algoritma kedua .
d. Agar pengerjaan penelitian ini tepat pada waktunya, maka beberapa penyesuaian perlu dilakukan pada aplikasi yang sedang dikembangkan dengan mengganti beberapa unsur penting dalam kriptografi dan menggantinya dengan unsur yang lebih sederhana.

2. METODE PENELITIAN

Metode yang digunakan untuk membuat aplikasi ini adalah metode *waterfall* dengan tujuan agar proses penelitian bisa berjalan dengan lancar :

a. Pertama, melakukan riset untuk mengetahui apa saja yang diperlukan untuk memulai penelitian. Serta memahami keamanan data yang diperlukan di SMA Sumbangsih dan algoritma yang cocok untuk digunakan dalam proses enkripsi dan dekripsi.
b. Kedua, setelah melakukan riset dan menentukan algoritma yang cocok untuk digunakan, maka berikutnya menganalisi sistem dan mengetahui arsitektur aplikasi dan kebutuhan yang diperlukan dalam pengembangan aplikasi.
c. Ketiga, pada tahap ini aplikasi yang dikembangkan telah dikembangkan ke dalam sebuah program kecil bernama *uni*t. Setiap *unit* yang dikembangkan akan diuji fungsinya untuk mencari kesalahan dan kekurangan dalam program.
d. Keempat, di dalam tahap ini semua *unit* yang telah diuji fungsinya akan diintegrasikan ke dalam aplikasi tersebut. Setelah semua *unit* sudah diintegrasikan akhirnya melakukan pengujian aplikasi untuk mencari kesalahan dan kekurangan lebih lanjut.
e. Kelima, Pada tahap akhir ini apikasi yang telah dibuat akan dijalankan dan melakukan pemeliharaan (*maintenance*) pada aplikasi tersebut. Tahap terakhir ini digunakan untuk memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya agar pengembangan aplikasi ini bisa menjadi lebih lancar dan cepat.

3. HASIL DAN PEMBAHASAN

3.1 Penyelesaian Masalah

Dari permasalahan yang teah dijelaskan, SMA Sumbangsih membutuhkan aplikasi keamanan *emai*l yang sederhana untuk keamanan data atau informasi pada *email*. Aplikasi ini akan menggunakan algoritma *Blowfish* dan *RC4* sebagai algoritma untuk memulai proses enkripsi dan dekripsi yang merupakan fungsi kriptografi pada aplikasi tersebut. Proses enkripsi yang merupakan proses pengamanan email ini akan dimulai dengan proses enkripsi algoritma *Blowfish* dan *RC4*, proses tersebut mengacak isi file dan pesan di dalam email sehingga tidak bisa terbaca agar kerahasiaan dan keamanan *file* dan pesan terjaga. *File* dan pesan yang sudah terenkripsi bisa dibaca kembali dengan menggunakan proses dekripsi dengan menggunakan proses dekripsi algoritma *Blowfish* dan *RC4*, file dan pesan yang terenkripsi akan didekripsi dengan proses tersebut agar isi pesan dan file yang ada dalam email bisa terbaca kembali.

3.2 Tampilan Layar

Tampilan layar menjadi sebuah aspek yang penting dalam menggunakan dan memahami cara kerja suatu program, dan juga agar para pengguna bisa merasa nyaman dan tidak mengalami kesulitan untuk menggunakan program tersebut.

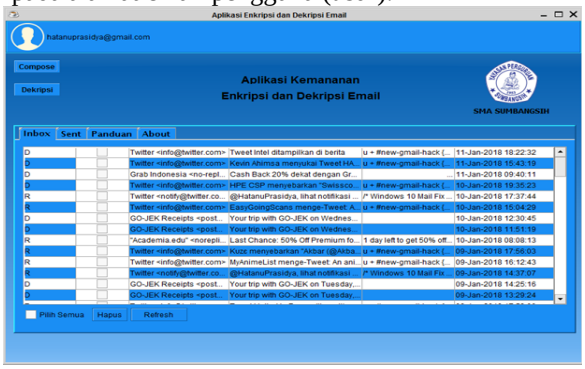
a. Tampilan Layar *Login*

Pada tampilan layar login terdapat dua inputan penting agar dapat masuk ke layar menu utama yaitu email dan password, akun yang bisa digunakan untuk login hanya akun *google mail* (*Gmail*). Bila pengguna mengalami kesulitan untuk *login* maka ada tombol Help agar memudahkan pengguna untuk melakukan *login*. Sedangkan tombol *Close* akan menutup aplikasi tersebut.

Gambar 1 : Tampilan Halaman *Login*

b. Tampilan Menu Utama

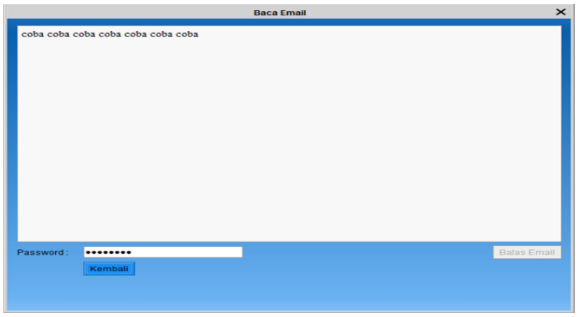
Pada tampilan form menu utama aplikasi, user dapat melihat *inbox* dan *send email* yang ada pada alamat email pengguna (*user*).



Gambar 2 : *Form* Menu Utama Aplikasi

c. Tampilan Email Setelah Didekrip

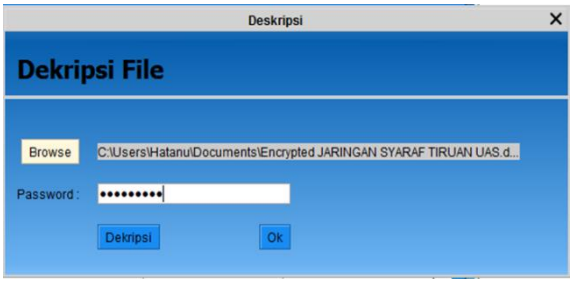
Pada tampilan pendekripan teks pesan yang di buka, setelah *password* yang di masukan benar



Gambar 3 : *Form Email* Setelah Didekrip Pesannya

d. Tampilan *Form* Deskripsi

Pada tampilan *form* dekrip file user dapat mendekrip file yang sudah di save pada pesan yang di baca. Berikut ini adalah tampilan layar form menu utama dekrip yang akan mendekrip file jika password yang dimasukan sama saat waktu di enkrip.



Gambar 4 : *Form* Dekripsi *File*

e. Tampilan Pesan Enkripsi

Hasil dari proses enkripsi akan muncul informasi seperti di bawah ini :

Gambar 5 : Tampilan Pesan Enkripsi *File*

f. Tampilan Pesan Dekripsi

Hasil dari proses dekripsi akan muncul in*Form* asi seperti di bawah ini :

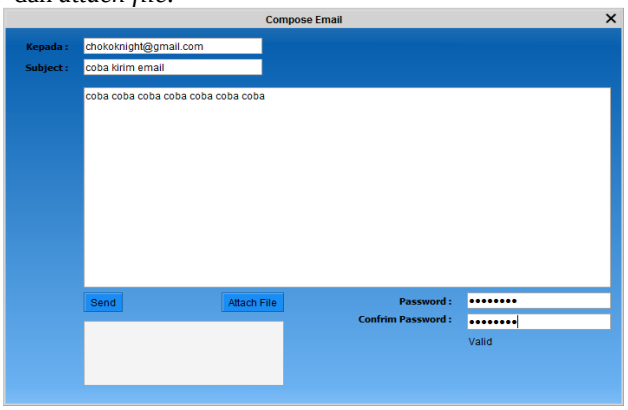
Gambar 6 : Tampilan Pesan Dekripsi *File*

3.3 Uji Coba Program

Uji coba pada aplikasi ini berguna untuk memperlihatkan cara kerja program untuk membantu pengguna mempelajari cara penggunaan apikasi tersebut tanpa ada kesulitan dan kesalahan.

a. Uji Coba Compose *Email*

Pada uji coba *compose Email* pengguna perlu mengisi enam inputan penting untuk memulai proses enkripsi pesan dan *file* dalam *email*. Input yang perlu diisi adalah input penerima, *subject*, isi pesan, *password*, *confirm password* dan *attach file*.

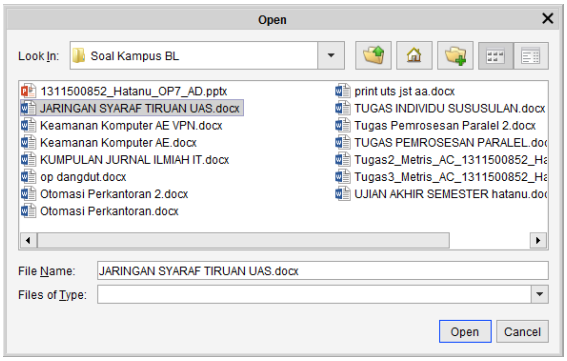


Gambar 7 : *Form* Compose *Email*

b. Uji Coba *Attachment File*

Uji coba ini akan menunjukkan cara memilih *file* yang akan dimasukkan ke dalam inputan *attach file*.

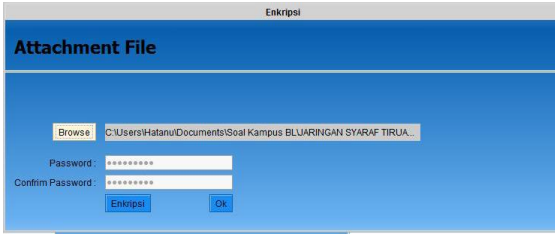
Gambar 8 : *Form Attachment File*



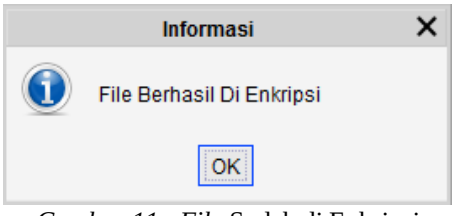
Gambar 9 : Membuka dan Memilih *File*

c. Uji Coba Enkripsi *File*

Tahap uji coba enkripsi *file* ini merupakan aspek penting untuk memulai pengamanan *file* yang akan dimasukkan ke dalam *email*.



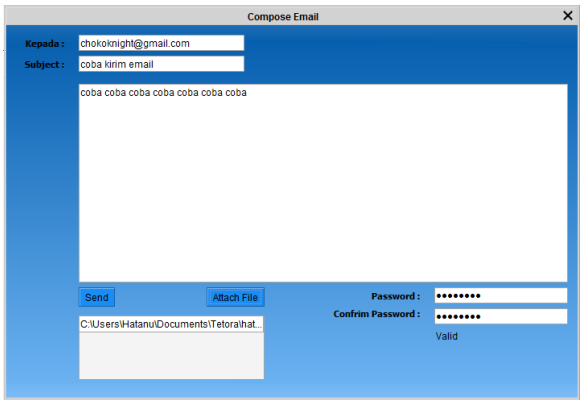
Gambar 10 : *File* Sudah Terpilih



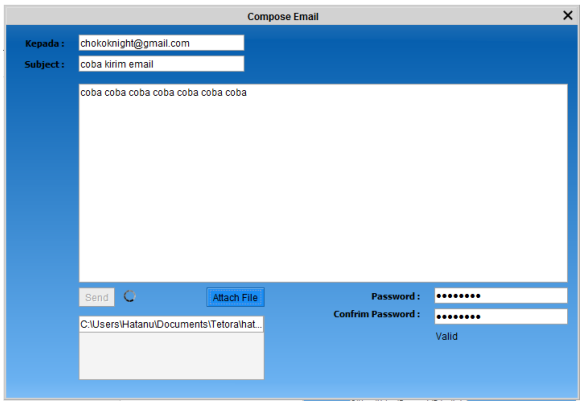
Gambar 11 : *File* Sudah di Enkripsi

d. Uji Coba Mengirim *Email*

Setelah semua uji coba sebelumnya sudah berhasil dicoba dan tidak ada kesalahan dan kesulitan, sekarang memasuki uji coba yang terakhir untuk mengrim *email* dan *file* yang sudah dienkripsi.



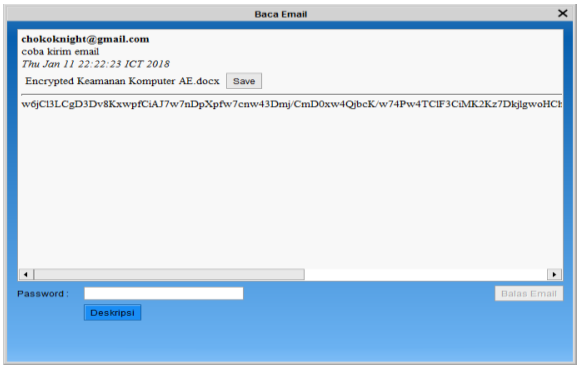
Gambar 12 : *Form Compose Email Send*



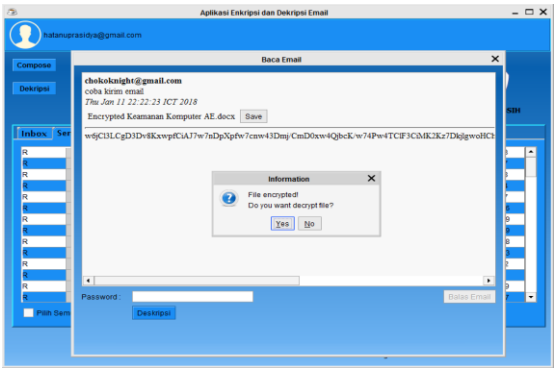
Gambar 13 : Proses Mengirim *Email*

e. Uji Coba Dekripsi *File*

Setelah uji coba enkripsi *file* untuk mengamankan pesan dan *File* di dalam *Email*, sekarang tinggal melakukan uji coba dekripsi *file* untuk membaca *file* yang sudah di enkripsi.



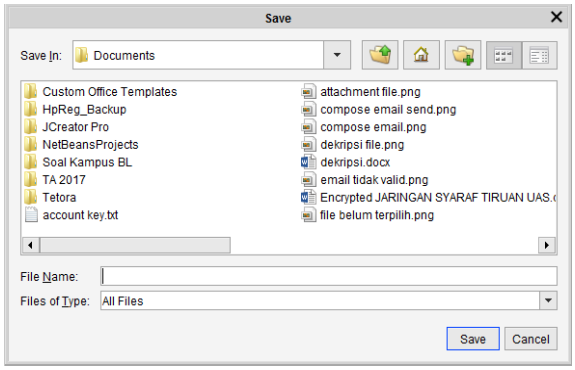
Gambar 14 : *Form Read Send Email*



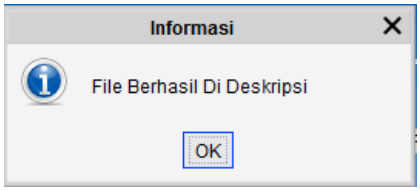
Gambar 15 : Save Attachment

Gambar 16 : *File* Dalam Keadaan Terenkripsi

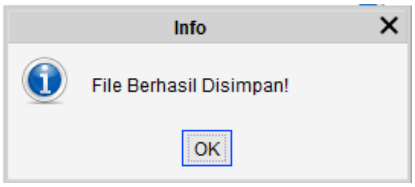
Gambar 17 : *Form* Dekripsi *File*



Gambar 18 : *Save File* Dekripsi



Gambar 19 : *File* Berhasil di Dekripsi



Gambar 20 : *File* Sudah Tersimpan

3.4 Evaluasi Program

Evaluasi pada aplikasi ini dilakukan untuk mengetahui kelebihan dan kekurangan yang terdapat pada aplikasi tersebut agar membantu pengembangan aplikasi antara lain :

a. Kelebihan Program

1) Tampilan aplikasi sederhana sehingga diharapkan mudah digunakan oleh *user*.
2) *File* hasil enkripsi nama *File* ditambahkan "Encrypted" dan *File* yang sudah di dekripsi akan ditambahkan "Decrypted" serta nama *File* ditentukan oleh user.
3) *File* hasil dekripsi tidak mengalami perubahan atau kerusakan dan dapat dibaca kembali oleh *user*.
4) Apabila kunci yang dimasukan salah pada proses dekripsi maka muncul message box peringatan "Password Salah" sehingga tidak akan terjadi proses dekripsi karena itu tidak akan membuat *user* menunggu lama.

b. Kekurangan Program

1) Aplikasi hanya bisa mengenkripsi *File* dengan filter berbasis ekstensi .txt, .doc, .docx, .xlsx, .xls dan .pdf .
2) Apabila ukuran *file* lebih besar maka waktu proses enkripsi yang dibutuhkan semakin lama.

4. KESIMPULAN

4.1 Kesimpulan

Berikut ini merupakan beberapa kesimpulan yang dapat ditarik:

a) Implementasi algoritma kriptografi *Blowfish* dan *RC4* (*Rivest Code* 4) dapat dilakukan dengan baik pada aplikasi pengamanan pesan *Email*.
b) Dengan aplikasi ini dapat meminimalisir kemungkinan pencurian dan manipulasi data oleh pihak yang tidak bertanggung jawab.
c) Data yang telah di enkripsi ketika dekripsi akan kembali ke ukuran semula tanpa mengalami perubahan

d) Dengan adanya aplikasi pengamanan *file* dokumen ini maka berbagi data menjadi lebih aman.
e) Dengan digunakannya dua metode maka *file* hasil enkripsi dapat lebih baik keamanannya.

4.2 Saran

Agar pengembangan lebih lanjut pada aplikasi enkripsi dan dekripsi dengan algoritma *Blowfish* dan *RC4* (*Rivest Code* 4) ada beberapa saran yang diberikan antara lain:

a) Diharapkan menggunakan spesifikasi hardware yang lebih tinggi agar proses enkripsi bisa lebih cepat.
b) Format data digital yang dapat dienkripsi diharapkan bisa lebih banyak lagi termasuk *file* gambar ekstensi .jpg, ,jpeg, .png, .gif, dan *file* musik ekstensi .mp3, .mp4.
c) Melakukan kombinasi dengan algoritma lain guna meningkatkan keamanan yang lebih baik.
d) Mengimplementasikan dengan bahasa pemprograman yang lain, berbasis web.

DAFTAR PUSTAKA

[1] Andri, M.Yuli. 2009. Implementasi Algoritma Kriptografi DES, RSA, dan Algoritma Kompresi LZW pada Berkas Digital. Skripsi. Medan, Indonesia: Universitas Sumatera Utara.

[2] Ariyus, Donny. 2006. Kriptografi Keamanan Data Dan Komunikasi, Yogyakarta: Graha Ilmu.

[3] Trianggana, Dimas Aulia & Herlina Latipa Sari. 2015. Analisis Perbandingan Kinerja Algoritma Blowfish Dan Algoritma Twofish Pada Proses Enkripsi Dan Dekripsi, Jurnal Pseudocode, Volume 2 Nomor 1, hh. 37-44.

[4] Fahlevi, Reza 2014, Perancangan Aplikasi *Email* – Recaiver Dengan Menerapkan Metode Secure Socket Layer, Jurnal Pelita In*Form* atika Budi Darma, Vol. VII(No.3), ISSN: 2301-9425, diakses 31 Juli 2016, <http://www.pelita-in*Form* atika.com/berkas/jurnal/25.%2520reza%25200fahle.pdf>.

[5] Haji, Wachyu Hari & Slamet Mulyono. 2012. Implementasi RC4 Stream Cipher Untuk Keamanan Basis Data, Seminar Nasional Aplikasi Teknologi In*Form* asi (ANATI 2012).

[6] Hermanto, Adi, Arman Rubi Fahlevi & Wayan Crisdiantoro. 2015. Aplikasi Keamanan Dokumen Menggunakan Algoritma Rivest Code 4 Dan Blowfish Berbasis Java, Jurnal Palcomtech. < http://news.palcomtech.com/wp-content/uploads/downloads/2015/12/Jurnal_Adi_Arman_Wayan_Aplikasi-Keamanan-Dokumen.pdf>.

[7] Kavi. "How *Email* Really Works." https://www.oasis-open.org/khelp/kmlm/user_help/html/how_*Email*_works.html (diakses tanggal 31 maret 2017).

[8] Mariyam, Siti. 2008. Aplikasi Sistem Pengaman Data Dengan Metode Enkripsi Menggunakan Algoritma RC4, jurnal dari http://ejournal.narotama.ac.id/.

[9] Menezes, J. Alfred, Paul C. Van Oorschot, Scott A. Vanstone, 1996, Handbook of Applied Cryptography, Boca Raton: CRC Press.

[10] Panjaitan, Yonathan Gani, Ajib Susanto, Wijanarto & Ibnu Utomo W.M . 2017. Enkriptor-Dekriptor Isi E-Mail Berbasis Android Dengan Algoritma Blowfish, Jurnal SIMETRIS, Vol 8 No 1. 193-200.

[11] Rifai, Rizal Yunan, Yuli Christyono & Imam Santoso. 2016. Implementasi Algoritma Kriptografi Rivest Code 4, Rivest Shamir Adleman, Dan Metode Steganografi Untuk Pengamanan Pesan Rahasia Pada Berkas Teks Digital, Jurnal Universitas Diponegoro Semarang, TRANSIENT, VOL.5, NO. 1.

[12] S. Sitinjak, Y. Fauziah, and Juwairah, "Aplikasi Kriptografi *File* Menggunakan," Seminar Nasional In*Form* atika 2010 (semnasIF 2010), pp. 78-86, 2010.

[13] Susanto. 2017. Implementasi Keamanan Data Menggunakan Algoritma Blowfish Pada Sistem In*Form* asi Koperasi Rias, Jurnal SIMETRIS, Vol 8 No 1. hh. 251-263.

[14] Trianggana, Dimas Aulia & Herlina Latipa Sari. 2015. Analisis Perbandingan Kinerja Algoritma Blowfish Dan Algoritma Twofish Pada Proses Enkripsi Dan Dekripsi, Jurnal Pseudocode, Volume 2 Nomor 1, hh. 37-44.

[15] Wasino, T. Puji Rahayu & Setiawan. 2012. Implementasi Steganografi Teknik End Of File Dengan Enkripsi Rijndael, Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012). hh. 150-157