Terbit online pada laman web jurnal: **https://journal.budiluhur.ac.id/index.php/bit**

JURNAL BIT
(Budi Luhur Information Technology)

Vol. 17 No. 2 (2020) 01 - 06 ISSN Media Elektronik: **2685-127X**

Aplikasi Pengamanan Data Email Menggunakan Algoritma Kriptografi XTEA Berbasis WEB

M. Anif[1)], Siswanto[2)], Basuki Hari Prasetyo[3)], Muhammad Fachri[4)], Gunawan Pria Utama[5)]

[1)2)3)4)] Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia
Jl. Ciledug Raya, Petukangan Utara, Kebayoran Lama Jakarta Selatan, 12260
E-mail : muhammad.anif@budiluhur.ac.id[1)], siswanto@budiluhur.ac.id[2)], haribhopal01@yahoo.com[3)]
fachribl@gmail.com[4)], gputama@gmail.com[5)]

Abstract

PT. Tanabe Indonesia is engaged in manufacturing in the health sector and uses cloud-based email and folder sharing facilities. In this case the company does not have a security system that can secure the contents of e-mail messages. In its implementation, the process of sending email on a network does not rule out being read by irresponsible people, such as intruders. This problem can be solved by creating an Email Data Security Application Using a web-based XTEA (Xtended Tiny Encryption Algorithm) Cryptographic Algorithm, so that email messages are not easily compromised. The application that is built also has additional facilities, namely the login process to be able to access it, so that only certain parties who have access rights can use this application. The programming language used in building this data security application is the web-based PHP programming language. From the test results, the file format will change to .sc format. and after going through the encrypt process it will change to be bigger with the average file size increasing by about 22.221 percent from the original file size before going through the encrypt process. From the test results after going through the decrypt process the average file size change will increase by 77.779% percent (22.221 percent reduction), which is the same as the average in the encrypt process.

Keywords: *XTEA cryptographic algorithms, data security, PHP, EMAIL*

Abstrak

PT. Tanabe Indonesia bergerak di bidang *manufacture* dalam bidang kesehatan dan menggunakan fasilitas email maupun *sharing folder* berbasis *cloud*. Dalam hal ini perusahaan tersebut belum mempunyai sjstem keamanan yang dapat mengamankan isi dari pesan email. Dalam implementasinya, proses pengiriman email di dalam suatu jaringan tidak menutup kemungkinan untuk dibaca oleh orang yang tidak bertanggung jawab, seperti penyusup. Masalah tersebut dapat diatasi dengan membuat Aplikasi Pengamanan Data Email Menggunakan Algoritma Kriptografi XTEA*(Xtended Tiny Encryption Algorithm)* berbasis web, agar pesan email tersebut tidak mudah dibobol. Aplikasi yang dibangun juga terdapat fasilitas tambahan, yaitu proses login untuk dapat mengaksesnya, sehingga hanya pihak tertentu yang memiliki hak akses yang dapat menggunakan aplikasi ini. Bahasa pemrograman yang digunakan dalam membangun aplikasi pengamanan data ini adalah bahasa pemrograman PHP yang berbasis web. Dari hasil uji coba, *format file* akan berubah menjadi *format .sc*. dan setelah melalui proses *encrypt* akan berubah menjadi lebih besar dengan rata-rata ukuran file bertambah sekitar 22,221 persen dari ukuran asli file sebelum melalui proses *encrypt.* Dari hasil uji coba setelah melalui proses *decrypt* rata-rata perubahan ukuran *file* akan bertambah sebesar 77,779% persen (berkurang 22,221 persen), sama dengan rata-rata pada proses *encrypt*.

Kata kunci: *algoritma kriptografi XTEA, pengamanan data,* PHP, EMAIL

1. Pendahuluan

PT. Tanabe Indonesia nergerak di bidang *manufacture* dalam bidang kesehatan, dimana menyediakan jasa/ layanan bahkan jual beli produk yang di miliki para pelanggannya. Dalam komunikasi baik internal maupun eksternal, PT. Tanabe Indonesia menggunakan fasilitas *email* maupun *sharing folder* berbasis *cloud*. Termasuk informasi penting yang menjadi perusahaan dikirim melalui *email* maupun *sharing folder* berbasis *cloud*. Berdasarkan permasalahan tersebut, akan dibangun aplikasi pengamanan *email* menggunakan metode kriptografi XTEA berbasis *web*. Sistem yang berjalan

Diterima Redaksi : 26-09-2020 | Selesai Revisi : 30-09-2020 | Diterbitkan Online : 30-09-2020

saat ini perusahaan belum mempunyai keamanan data yang tepat dalam melakukan komunikasi antar karyawan melalui *email*. Dimana hal tersebut dimaksudkan agar pesan *email* penting yang menjadi rahasia perusahaan dapat diamankan, sehingga pihak lain tidak dapat mengetahui informasi yang terdapat pada data penting tersebut.

XTEA adalah Salah satu algoritma yang dapat digunakan untuk melakukan enkripsi data sehingga data asli hanya dapat dibaca oleh seseorang yang memiliki *key* enkripsi tersehut. XTEA beroperasi dalam ukuran blok 64 *bit* dan panjang kunci (*key*) 128 bit. XTEA berbasiskan jaringan *Feistel* dan memiliki 32 putaran. Pada XTEA, pada ronde ganjil digunakan k[sum & 3], sedangkan pada ronde genap digunakan k[sum >> 11 & 3]. XTEA menggunakan angkat delta yang didapatkan dari rumus *golden number*, yaitu delta = $(\sqrt{5} - 1)\ 2^{31}$ = 0x9E3779B9 = -1640531527 [1].

Langkah awal sebelum melakukan proses enkripsi maupun dekripsi yaitu melakukan proses pembangkit kunci, sehingga menghasilkan *subkey* yang nantinya akan digunakan pada proses enkripsi maupun dekripsi [2].

Implementasi algoritma kriptografi XXTEA meningkatkan sekuritas pada aplikasi Online Test di SMK Immanuel Pontianak dengan diujikan pada server secara online menggunakan *sniffing, brute force* dan *SQL Injection, karena a*lgoritma kriptografi XXTEA menggunakan kelipatan blok 32 bit dengan minimum 64 bit, sehingga melalui perhitungan matematis dan pengujian menggunakan *brute force* memungkinkan untuk membobol enkripsi yang telah diimplementasi pada aplikasi, namun memerlukan waktu 4,2 x 10^9 tahun untuk data 32 bit untuk mencoba semua kemungkinan yang mungkin menjadi hasil dari enkripsi yang ter-capture melalui sniffing [3].

Pada enkripsi dan dekripsi file, file yang akan dienkripsi ialah file-file yang terdapat dalam sistem komputer seperti file Microsoft Office, file MP3, file PDF, dan file gambar. File yang telah dienkripsi akan berekstensi aes dan untuk melihat hasil dari enkripsi file tersebut dapat dilihat menggunakan notepad [4].

Fle citra (gambar) yang telah disisipkan atau disembunyikannya sebuah pesan atau file rahasia berupa file kompresi setelah dilakukan proses penyembunyian (*hidden*) file citra (gambar) tidak mengalami banyak perubahan yaitu citra (gambar) yang dihasilkan terlihat masih sama dengan citra (gambar) aslinya, hanya berbeda pada ukurannya yaitu ukuran filenya akan lebih besar dibandingkan dengan file citra (gambar) aslinya atau sebelum dilakukan proses *hidden*. Hal itu dikarenakan file gambar tersebut disisipkan atau disembunyikan sebuah pesan atau file rahasia, dimana pada file gambar yang asli dengan ukuran yang asli tersebut akan bertambah dengan ukuran pesan atau file rahasia tersebut [5].

Hasil dari pengujian kriptografi dengan algoritma kriptografi TEA (*Tiny Encryption Algorithm*) ini, data dapat diamankan untuk menghindari serangan cryptanalysis. Rata-rata ukuran file yang telah melalui proses *encrypt* bertambah sekitar 33,29512 persen dari ukuran asli file sebelum melalui proses encrypt. Rata-rata perubahan ukuran file yang telah melalui proses *decrypt* akan berkurang sebesar 25.0231 persen [6].

Hasil penelitian G. Qian, S. Sural, Y. Gu, and S. Pramanik, *Similarity between Euclidean and cosine angle distance for nearest neighbor queries,* membuat sistem yang berfokus pada implementasi FPGA ringan algoritma kriptografi Enkripsi Algoritma TEA untuk beradaptasi dengan banyak kendala real time seperti memori, kehilangan data dan biaya rendah. Skema yang diusulkan menggun,akan *Linear Feedback Shift Register* untuk menghasilkan kunci acak sehingga lebih aman untuk transfer informasi sensitif di banyak aplikasi *real time* [7].

Hasil penelitian M. Shoeb and V. K. Gupta, *A Crypt Analysis Of The Tiny Encryption Algorithm In Key Generation,* mengimplementasikan algoritma enkripsi yang digunakan untuk keamanan lebih komunikasi nirkabel, tetapi mengamankan data juga mengkonsumsi sumber daya. Faktor penting utama yang perlu dipertimbangkan ketika merancang sistem kriptografi adalah kinerja, kecepatan, ukuran, dan keamanan. *Tiny Encryption Algorithm* (TEA), dan eXtended TEA (XTEA) adalah contoh dari algoritma kriptografi. *Tiny Encryption Algorithm* (TEA) adalah algoritma kriptografi yang dirancang untuk meminimalkan pemakaian memori dan memaksimalkan kecepatan. Ini adalah jenis cipher feistel yang menggunakan operasi dari campuran (*orthogonal*) kelompok aljabar [8].

Hasil dari pengujian aplikasi chatting berbasis web dengan menggunakan algoritma kriptografi X-TEA ini berdasarkan file yang pernah dicoba lama eksekusi tergantung pada ukuran file, agar transaksi cepat dijalankan dengan cepat file yang dikirim dibawah 1MB (1024kb). Dan untuk rata rata penambahan ukuran file setelah di enkrip adalah sekitar 44% dan pengurangan nya 44% waktu yang diperlukan unruk sekali enkrip tergantung ukuran file [9].

Karakter yang dihasilkan pada transformasi Base64 ini terdiri dari A..Z, a..z, dan 0..9, serta ditambahkan dengan dua karakter terakhir yang bersimbol + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data binary atau istilahnya disebut dengan pengisi pas. Karakter simbol yang akan dihasilkan tergantung dari proses algoritma yang berjalan [10].

2. Metode Penelitian

Metode penellitian yang digunakan dalam penelitian ini, langkah-langkah sebagai berikut:

2.1 Analisa Masalah

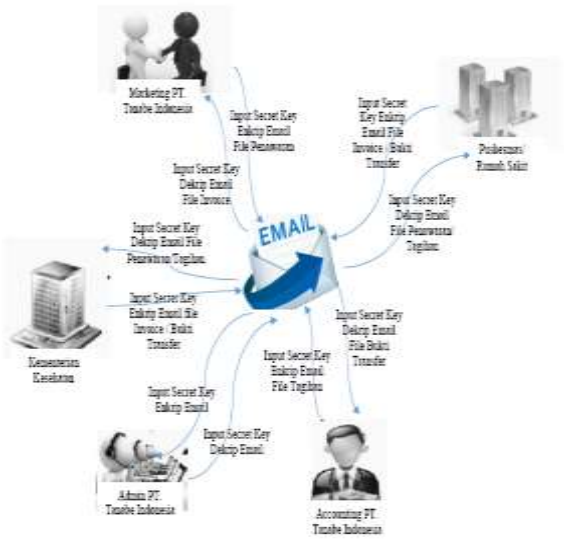
Analisa masalah dilakukan dengan penelitian langsung ke PT. Tanabe Indonesia yang diteliti untuk mendapatkan data email dan informasi yang harus diamankan serta masalah yang sering terjadi selam proses transfer file yang berisikan data penting.

2.2 Analisa Algoritma

Mempelajari cara kerja algoritma XTEA [1][9] dan Base64. Algoritma XTEA terdiri dari 3 fungsi, yaitu Proses Pembangkit Kunci, Proses Enkripsi, dan ProsesDekripsi. Pada proses pembangkit kunci, string darikunci diubah menjadi nilai desimal dari ASCIIyang dibagi menjadi 4 blok 32 bit. Lalu pada proses enkripsi, sama halnya dengan proses pembangkitkunci dimana string dari *plaintext* diubah menjadinilai desimal dari ASCII yang dibagi menjadi perblok 64 bit (dipecah kembali menjadi per 32 bit V0& V1), yang selanjutnya dilakukan perhitunganberikut sebanyak 32 perulangan (Delta =9E3779B9): Sum = 0 V0 += ((V1 « 4) XOR (V1 » 5) + V1) XOR (Sum + (S[Sum AND 3])), Sum += Delta, V1 += ((V0 « 4) XOR (V0 » 5) + V0) XOR (Sum + (S[Sum » 11 AND 3])), Dan untuk proses dekripsi, sama halnya dengan proses enkripsi namun perbedaannya terletak pada perhitungan berikut yang dilakukan sebanyak 32 perulangan (Delta = 9E3779B9): Sum = Delta * 32 V1 -= ((V0 « 4) XOR (V0 » 5) + V0) XOR (Sum + (S[Sum » 11 AND 3])), Sum -= Delta, V0 -= ((V1 « 4) XOR (V1 » 5) + V1) XOR (Sum +(S[Sum AND 3])), 3.2 Algorimta Base64 Langkah-langkah dari algoritma encoding Base64 sebagai berikut : a. String bytes di pecah menjadi per-3 bytes. b. Gabungkan 3 bytes menjadi 24 bit.. c. Simpan 24 bit di-buffer lalu dipecah-pecah menjadi 6 bit, maka akan menghasilkan 4 pecahan. d. Masing-masing pecahan diubah ke dalam nilai decimal. e. Gunakan nilai-nilai desimal tersebut menjadi indeks untuk memilih karakter penyusun dari base64 dan mulai dari 0 maksimal adalah 63 atau indeks ke 64. Sampai akhir string bytes yang mau dikonversikan. Jika ternyata dalam proses encoding terdapat sisa pembagi, maka tambahkan sebagai penggenap sisa tersebut karakter = (sama dengan). Teknik decodingBase64 sebenarnya sederhana, jika ada satu (string) bytes yang akan disandikan ke Base64 maka caranya adalah sebagai berikut : a. Pecah string bytes tersebut ke per-4 bytes. b. Gabungkan 4 bytes menjadi 24 bit. Dengan catatan 1 bytes = 6 bit, sehingga 4 x 8 = 24 bit. c. Lalu 24bit yang disimpan di-buffer (disatukan) dipecah-pecah menjadi 8 bit, maka akan menghasilkan 3 pecahan. d. Jika terdapat karakter = (sama dengan) maka karakter ini dihilangkan, karena merupakan padding. e. Masing-masing pecahan diubah ke dalam nilai decimal. f. Terakhir, jadikan nilai-nilai desimal tersebut menjadi indeks kode ASCII dan maksimal adalah 256 atau indeks ke-256. Dan seterusnya sampai akhir stringbytes yang mau dikonversikan,

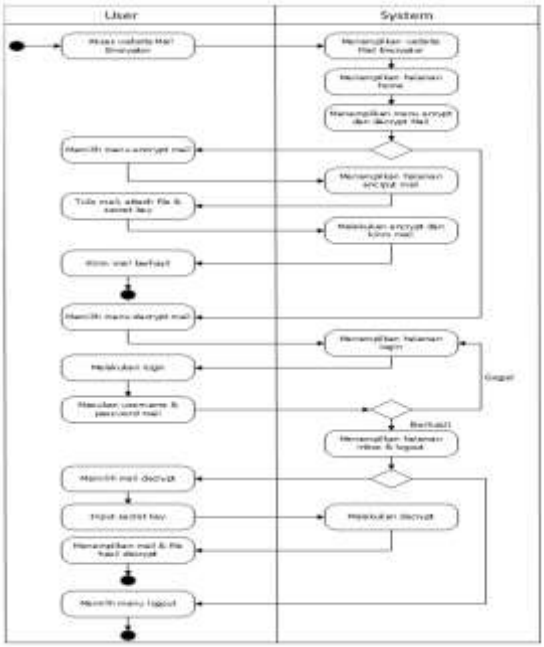
2.3 Mendesain Aplikasi

Gambaran aktifitas mengamankan data email dapat dimodelkan dalam *rich picture* seperti gambar 1, desain activity diagram dan user interface aplikasi pengamanan data yang akan digunakan untuk mengamankan pesan email



Gambar 1. ***Rich Picture Mrngamankan Data Email***

Gambar 2. berikut ini merupakan *activity diagram* bagaimana *user* dan sistem bekerja pada aplikasi mengamankan data email tersebut.



Gambar 2. ***Activity Diagram*** **Aplikasi Mengamankan Data**

Tahap awal dalam perancangan adalah halaman *menu utama*, yang berisi halaman untuk memilih memilih menu *encryption mail* atau *decryption mail*, seperti yang terlihat pada gambar 3.

Gambar 3. Tampilan layar *form* desain halaman *menu utama*

Pada menu *encryption mail* akan langsung ditampilkan menu untuk melakukan *input* pesan dan *upload file* seperti yang terlihat pada gambar 4.

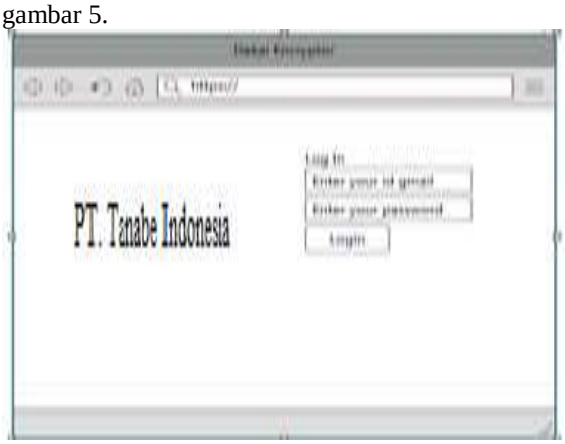
Gambar 4. Tampilan layar *form* desain *menu encryption mail*

Pada tampilan halaman *encrypt* ini ada beberapa *menu* yang berfungsi sebagai berikut :

1) *Home*, untuk kembali ke menu utama (*home).*
2) *Gmail ID*, kolom yang berfungsi untuk memasukan *user id Gmail.*
3) *Password*, kolom yang berfungsi untuk memasukan *password Gmail* yang telah dimasukan sebelumnya.
4) *To*, kolom berfungsi untuk memasukkan alamat *gmail* yang dituju.
5) *Subject*, kolom yang berfungsi untuk memasukan *subject email* yang akan dikirim.
6) *Choose file*, tombol yang berfungsi untuk memiih *file* yang akan diunggah.
7) *Secret Key*, kolom yang berfungsi untuk memasukan *secret key.*
8) *Send*, tombol *send* berfungsi untuk mengirimkan pesan sekaligus untuk proses enkripsi.

Pada saat *user* hendak mengirimkan pesan dengan enkripsi maka diperlukan untuk memasukan *secret key* yang terdiri dari 16 character dan *case sensitif*, diwajibkan bagi *user* untuk mengingat *secret key* yang dimasukan, karena nantinya *secret key* ini akan digunakan kembali saat penerima *email* hendak membuka *email* ini kembali.

Pada halaman *decryption mail* akan langsung ditampilkan halaman *login*, dihalaman ini user mengisi *username* dan *password Gmail* untuk proses mengakses *inbox* di *gmail*, seperti yang terlihat pada gambar 5.



Gambar 5. Tampilan layar *form* desain menu *mail decryption*

Setelah *user* memasukan *username* dan *password* denga benar maka akan masuk ke halaman *inbox*. Pada halaman inbox ini, akan memuat pesan-pesan enkripsi yang diterima, selain itu ada menu *logout* untuk keluar dari halaman *inbox*. Tampilan layar form desain halaman *inbox* dapat dilihat pada gambar 6.

Gambar 6. Tampilan layar *form* halaman *inbox*

Pilih dan klik pesan yang akan di *decrypt*. Akan muncul *pop up* atau halaman *web* untuk memasukan *secret key* 16 karakter untuk melakukan *decrypt* dan juga terdapat tombol *submit* untuk melakukan proses *decrypt*. Berikut di gambar 7 adalah tampilan halaman web untuk memasukan *secret key*

Gambar 7. Tampilan layar *form* halaman memasukan *secret key*

2.4 Membuat program Aplikasi

Membuat program aplikasi pengamanan data yang akan digunakan untuk mengamankan pesan email dengan bahasa pemrograman PHP dan mengelola file log proses enkrip dan dekrip email dengan MySQl

2.5 Ujicoba Program Aplikasi

ujicoba program aplikasi dengan mencoba panjang pesan email yang kecil dan yang besar ukuran pesannya.

3. Hasil Dan Pembahasan

Pada pembahasan ini, dilakukan pengujian aplikasi yang telah dibangun dengan menggunakan *file*. Hasil dari pengujian yang dilakukan, *file* tersebut berhasil dilakukan proses pengamanan data, sehingga pihak lain tidak dapat melihat isi dari data yang sesungguhnya. *File* yang akan dilakukan proses *encrypt* dapat dilihat pada gambar 8.

Gambar 8: Tampilan file sebelum dan setelah proses *encrypt*

File hasil *encrypt* dapat dilihat pada gambar 9,

Gambar 9. Tampilan file setelah proses *encrypt*

Berikut adalah hasil pengujian dari beberapa file dengan format yang berbeda-beda didapat sebagai berikut :

Tabel 3: Tabel Hasil Uji Coba Proses *Encrypt*

No	Nama File	Sebelum *Encrypt*		Ukuran file sc (bytes)
		Format File	Ukuran File (bytes)	
1	File penawaran	**xls**	**147.418**	**262.104**
2	File invoice Kemenkes	**pdf**	**6.904.704**	**12.275.048**
3	File tagihan Kemenkes	**pdf**	**575.718**	**1.023.420**
4	File bukti transfer Kemenkes	**png**	**272.028**	**483.628**
5	File laporan keuangan	**xls**	**147.419**	**262.104**
6	File invoice Puskesmas	**pdf**	**186.283**	**331.196**
7	File tagihan Puskesmas	**pdf**	**186.272**	**331.168**
8	File bukti transfer Puskesmas	**pdf**	**23.224**	**41.304**
9	File Laporan pemesanan	**doc**	**63.070**	**112.144**
10	File invoice Rumah Sakit	**pdf**	**258.952**	**460.376**

Dari hasil uji coba, *format file* akan berubah menjadi *format .sc.* dan untuk ukuran *file* setelah melalui proses *encrypt* akan berubah menjadi lebih besar dengan rata-rata ukuran file yang telah melalui proses *encrypt* bertambah sekitar 22,221 persen dari ukuran asli file sebelum melalui proses *encrypt*.

Tabel 4: Tabel hasil uji coba proses *decrypt*

No	Nama File	Ukuran File .sc (bytes)	Setelah *Decrypt*	
			Format File	Ukuran File (bytes)
1	File penawaran	**262.104**	**xls**	**147.418**
2	File invoice Kemenkes	**12.275.048**	**pdf**	**6.904.704**
3	File tagihan Kemenkes	**1.023.420**	**pdf**	**575.718**
4	File bukti transfer Kemenkes	**483.628**	**png**	**272.028**
5	File laporan keuangan	**262.104**	**xls**	**147.419**
6	File invoice Puskesmas	**331.196**	**pdf**	**186.283**
7	File tagihan Puskesmas	**331.168**	**pdf**	**186.272**
8	File bukti transfer Puskesmas	**41.304**	**pdf**	**23.224**
9	File Laporan pemesanan	**112.144**	**doc**	**63.070**
10	File invoice Rumah Sakit	**460.376**	**pdf**	**258.952**

Dari hasil uji coba *format file .sc* akan berubah menjadi *format* awal sebelum *file* dilakukan proses *encrypt*, ukuran *file* setelah melalui proses *decrypt* akan berubah menjadi ukuran awal *file* sebelum *encrypt.* Rata-rata perubahan ukuran *file* akan bertambah sebesar 77,779% persen (berkurang 22,221 persen), sama dengan rata-rata pada proses *encrypt*.

Tabel 5: Tabel Perbandingan Proses *Encrypt*

No	Nama File	Format File	Lama Proses *Encrypt* (MS)		
			Program (sc)	Twofish (tf)	TEA (tea)
1	File penawaran	**xls**	**84.89**	**35.9**	**52.53**
2	File invoice kemenkes	**pdf**	**5269.87**	**1977.92**	**3357.04**
3	File tagihan kemenkes	**pdf**	**2756.97**	**1099.41**	**1786.53**
4	File bukti transfer kemenkes	**png**	**39520.6**	**15497.91**	**25027.02**
5	File laporan keuangan	**xls**	**32726.32**	**12339.57**	**20379.3**

Dari hasil uji coba perbandingan kecepatan *encrypt*, untuk kecepatan pada proses *encrypt* berbeda, pada algoritma program Aplikasi ini paling lama dikarenakan algoritma TEA, dan untuk kecepatan *encrypt* pada TEA *round* yang dipakai TEA adalah 32.

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan serta uji coba aplikasi dapat disimpulkan bahwa data dapat diamankan dengan baik dengan algoritma kriptografi XTEA, data tidak dapat dibuka oleh pihak yang tidak berhak yang tidak memiliki kunci untuk *decrypt file.*

Dari hasil uji coba, *format file* akan berubah menjadi *format .sc*. dan untuk ukuran *file* setelah melalui proses *encrypt* akan berubah menjadi lebih besar dengan rata-rata ukuran file yang telah melalui proses *encrypt* bertambah sekitar 22,221 persen dari ukuran asli file sebelum melalui proses *encrypt.* Dari hasil uji coba *format file .sc* akan berubah menjadi *format* awal sebelum *file* dilakukan proses *encrypt*, ukuran *file* setelah melalui proses *decrypt* akan berubah menjadi ukuran awal *file* sebelum *encrypt.* Rata-rata perubahan ukuran *file* akan bertambah sebesar 77,779% persen (berkurang 22,221 persen), sama dengan rata-rata pada proses *encrypt*.

Adapun masukan dan saran agar dapat meningkatkan performansi aplikasi yang telah dibangun, serta sebagai perbaikan dan pengembangan agar program dapat melakukan *encrypt* dan *decrypt* dengan waktu yang lebih singkat untuk ukuran *file* yang besar, agar dapat melakukan proses *encrypt* dan *decrypt* pada beberapa *file* sekaligus dalam waktu yang bersamaan untuk alasan efisiensi dan juga agar dapat melakukan kompresi data yang di *encrypt*, sehingga ukuran data tidak akan terlalu banyak berubah dibandingkan dengan ukuran asli data.

5. Daftar Pustaka

[1] Jiazhe, Chen, dkk. 2011. Impossible Different Cryptanalysis of the lightweight Block CiphersTEA, XTEA and HIGHT, International Journal on Cryptology in Africa , ISSN : 2302 – 7339, Vol. 10 No.1. Tersedia Online di DOI: 10.1007/978-3-642-31410-0_8.

[2] Maruf, Fathir, dkk. 2015. Merging of Vignere Cipher With Extea Block Chiper To Encryption Digital Document, International Journal of Computer application, ISSN : 0975-8887, Vol. 132 No.1. December 2015.

[3] Yuricha, Tursina, Helfi N., 2017. Implementasi Algoritma Kriptografi XXTEA untuk Enkripsi dan Dekripsi Query Database pada Aplikasi OnlineTest (StudiKasus:SMK Immanuel Pontianak), Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol. 5, No. 1, (2017), pp. 42-46.

[4] Angga A. P. , Desi N., 2018. Rancangan Aplikasi Pengamanan Data Dengan Algoritma Advanced Encyption Standard (AES), Jurnal Teknik Informatika Vol 11 No. 2, Oktober 2018, pp. 177-186 p-ISSN 1979-9160, e-ISSN 2549-7901, Tersedia Online di https://dx.doi.org/10.15408/jti.v11i2.7811.

[5] Fresly N. P. , Indah F. A. , Awang H. K., 2015. Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard, Jurnal Informatika Mulawarman Vol. 10 No. 1 Februari 2015, pp. 20-31.

[6] Siswanto, M. Anif, dan Windu G., 2018. Penerapan Algoritma Kriptografi TEA Dan Base64 Untuk Mengamankan Email. Jurnal ELTIKOM, Vol. 2 No. 1, Juni 2018, pp. 34-41 ISSN 2598-3245 (Print), ISSN 2598-3288 (Online) Tersedia Online di http://eltikom.poliban.ac.id.

[7] G. Qian, S. Sural, Y. Gu, and S. Pramanik, 2004,. Similarity between Euclidean and cosine angle distance for nearest neighbor queries, in Proceedings of the 2004 ACM symposium on Applied computing - SAC ’04, 2004, pp. 12-32.

[8] M. Shoeb and V. K. Gupta, 2013. A Crypt Analysis Of The Tiny Encryption Algorithm In Key Generation, Int. J. Comput. Technol., vol. 1, no. 38, 2013.

[9] Hafidz S., Siswanto, 2018. Implementasi Algoritma Extended Tiny Encryption Algorthm Dan Base64 Untuk Mengamankan Aplikasi Chatting Berbasis Web, Jurnal Skanika, Volume 1 No. 3 Juli 2018. Pp. 1237-1241.

[10] *Azlin Azlin, Fithriah Musadat, Jabal Nur*. 2018, APLIKASI KRIPTOGRAFI KEAMANAN DATA MENGGUNAKAN ALGORITMA BASE64, Jurnal Informatika, Volume 7, No.2, Desember2018ISSN Online 2528-0090, pp. 1-5 diakses 20 Maret 2020, Tersedia Online di http://ejournal.unidayan.ac.id/index.php/JIU/article/view/55/105