IMPLEMENTASI ALGORITMA EXTENDED TINY ENCRYPTION ALGORTHM DAN BASE64 UNTUK MENGAMANKAN APLIKASI CHATTING BERBASIS WEB

Hafidz Shidiq[1)], **Siswanto**[2)]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail : hafidz.shidiq24@gmail.com[1)], siswanto@budiluhur.ac.id[2)]

ABSTRAK

Aplikasi chat pengamanan data ini dirancang untuk mengamankan data penting yang akan dikirm ke cabang cabang kantor pada PT. Bank Nationalnobu terutama data transaksi yang bersifat rahasia. Data yang dikirm adalah data transaksi nasabah yang rahasia, maka keamanan data tersebut menjadi sangat rentan terhadap pencurian dan manipulasi data dari berbagai pihak yang tidak bertanggung jawab mengingat sering juga data tersecibut dikirimkan menggunakan fasilitas chat. Seiring perkembangan teknologi untuk berkomunikasi maka perusahaan membutuhkan media komunikasi yang realtime untuk memudahkan transaksi agar cepat dijalankan, sehingga dibutuhkan aplikasi chatting berbasis web yang dapat diterapkan untuk mengonfirmasi transaksi nasabah cabang asal agar dapat dengan cepat dijalankan. Aplikasi ini dibuat menggunakan Bahasa pemograman PHP dan untuk menyimpan data menggunakan MYSQL sebagai database. Untuk meningkatkan keamanan komunikasi yang terjadi ketika pengguna menggunakan aplikasi chatting, maka dibutuhkan metode keamanan, yaitu dengan menggunakan algoritma kriptografi X-TEA (Extended Tiny Encryption Algorithm). Penelitian ini bertujuan untuk membuat aplikasi chatting yang mampu mengenkripsi dan dekripsi file text dan gambar yang dikirim oleh pengguna, sehingga komunikasi yang berlangsung aman dan tidak dapat dibaca oleh orang yang tidak berhak. Hasil dari pengujian aplikasi ini dapat disimpulkan bahwa 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.

Kata kunci : Kriptografi, X-TEA, Enkripsi, Dekripsi, Plaintext, *Ciphertext*, Aplikasi *Chatting*

1. PENDAHULUAN

PT. Bank Nationalnobu adalah perusahaan yang bergerak di bidang perbankan, banyak data nasabah yang sifatnya rahasia salah satunya adalah transaksi nasabah yang nominalnya besar. Transaski dengan nominal besar ini akan dikonfirmasi ke cabang pembuka rekening dengan cara dokumen transaksi dikirim melalui email. Sampai saat ini belum memiliki fasilitas keamanan berbasis kriptografi dan juga fasilitas pengamanan konten *email* sehingga masih rentan terhadap dengan pencurian data, manipulasi data atau penyadapan *email*. *Email* pun mempunyai kelemahan salah satunya adalah bahwa tidak semua orang membaca *email* setiap hari. Sehingga ada kemungkinan *email* terlambat dibalas dan itu akan membuat transaksi nasabah menjadi terlambat.

Demi keamanan dan kecepatan data yang dikrim, oleh karena itu untuk mengambil tema keamanan data dengan dibuatnya aplikasi *chatting* dengan diberi keamanan menggunakan enkripsi algoritma X-TEA (*Extended Tiny Encrypted Algorthm*) dan *Base64*. Dimana hal tersebut dimaksud agar data penting yang manjadi rahasia perusahaan dapat diamankan, sehingga pihak lain tidak dapat mengetahui informasi yang terdapat pada data penting tersebut. Dengan diterapkannya aplikasi pengamanan data menggunakan metode kriptografi XTEA (*Extended Tiny EncryptedAlgorthm*) dan *Base64* berbasis web pada PT. Bank Nationalnobu, sehingga didapatkan hasil data yang lebih aman dan tahan terhadap serangan *cryptanalysis* (pemecahan enkripsi).

Pada PT. Bank Nationalnobu sering mengirimkan data rahasia transaksi nasabah seperti yang dijelaskan sebelumnya.

Oleh karena itu diperlukan cara untuk mengamankan data transaksi yang sebelumnya dapat dibaca oleh semua orang, sehingga hanya orang yang berhak saja yang dapat membacanya dengan menggunakan aplikasi encrypt yang telah dibuat.

Sesuai dengan permasalahan yang telah dirumuskan sebelumnya, tujuan yang ingin dicapai dalam penelitian ini adalah untuk membangun aplikasi chat dengan berbasis web agar mudah di akses oleh karyawan PT. Bank Nationalnobudengan membuat aplikasi keamanan data dengan algoritma kriptografi X-TEA (Extended Tiny Encrypted Algorthm) dan Base64 untuk menghindari dari aksi-aksi peretasan dan pencurian data.

Adapun metodologi yang digunakan adalah salah satunya metoda *waterfall* yaituyang harus di selesaikan secara tertib, jadi dalam mengembangkan aplikasi, pengembang harus membuat requirement,

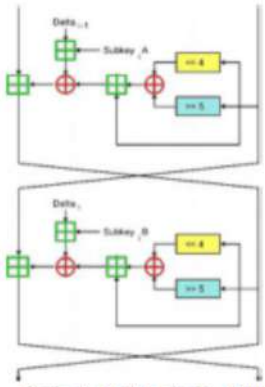
design, implement, develop, maintenance. setiap proses tersebut harus dikerjakan secara tertib, jadi jika ada kebutuhan tambahan maka harus menunggu sampai tahap akhir.

Percakapan yang dilakukan bisa berupa teks, suara, atau semuanya digabungkan antara teks, suara dan video [5].

Jenis – jenis strategi keamanan informasi sebagai berikut : *Physical security,* Personal *security*, *Operation security, Communication security, dan Network Security.*

Block Cipher menggunakan beberapa fungsi matematika, di antaranya fungsi permutasi dan fungsi substitusi, sehingga konfusi (*confusion*) dan difusi (*diffusion*) pada *blockcipher* dapat terpenuhi [10].

Yang telah mereka ciptakan pada tahun 1994. XTEA merupakan algoritma kriptografi *blockcipher* yang 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 S[Sum & 3], sedangkan pada rondegenap digunakan S[Sum >> 11 & 3] [3]. XTEA menggunakan angkat delta yang didapatkan dari rumus *goldennumber*, yaitu delta = $(\sqrt{5} - 1)\ 2^{31}$ = 0x9E3779B9. Algoritma XTEA terdiri dari tiga fungsi, yaitu Proses Pembangkit Kunci, Proses Enkripsi, dan Proses Dekripsi.



Gambar 1 : Struktur algoritma XTEA

Hal ini untuk memastikan bahwa data tetap utuh tanpa perubahan selama pengiriman. Penggunaan lain *encodingBase64* adalah untuk melakukan *obfuscation* atau pengacakan data. Skema enkripsi *Base64* biasanya juga digunakan ketika diperlukan sandi terhadap data *biner* yang didesain untuk menangani data berbentuk teks, hal ini ditujukan untuk menjaga data selama pengiriman ke suatu server. 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 [6].[7]

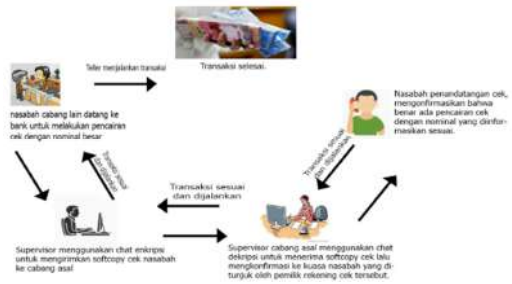
2. METODE PENELETIAN

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

a. Analisa Kebutuhan dilakukan dengan penelitian langsung ke PT. Bank Nationalnobu yang diteliti untuk mendapatkan data dan informasi yang harus diamankan serta masalah yang sering terjadi selam proses transfer file yang berisikan data penting.
b. Mempelajari cara kerja algoritma XTEA dan Base64.
c. Mendesain serta memodelkan algoritma dan user interface aplikasi chatting yang akan digunakan untuk mengamankan pesan chat.
d. Membuat program dengan bahasa pemrograman PHP dan mengelola file log proses enkrip dan dekrip chat dengan MySQl
e. Ujicoba Program dengan mencoba panjang pesan chat yang kecil dan yang besar ukuran pesannya.

3. HASIL DAN PEMBAHASAN

Berdasarkan permasalahan yang telah dijabarkan, apabila aplikasi kriptografi yang dibangun digambarkan *richpicture* untuk menyeselaikan masalah tersebut, maka dapat dilihat seperti pada gambar 2 .



Gambar 2: Rich Picture Penyelesaian Masalah

Apabila proses aplikasi kriptografi yang dibangun digambarkan dalam bentuk diagram aktifitas atau *Activity diagram*, maka proses dapat dilihat seperti pada gambar 3.:

Gambar 3 : Activity Diagram system kemanan data

3.1 Algoritma XTEA

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 :

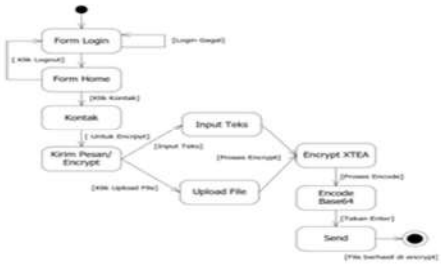
a. Pecah string bytes tersebut ke per-4 *bytes*.
b. Gabungkan 4 *bytes* menjadi 24 *bit*. Dengan catatan 1 *bytes* = 6 *bit*, sehingga 4 x 8 = 24 *bit*.
c. Lalu *24bit* yang disimpan di-*buffer* (disatukan) dipecah-pecah menjadi 8 *bit*, maka akan menghasilkan 3 pecahan.
d. Jika terdapat karakter = (sama 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

3.3 Program

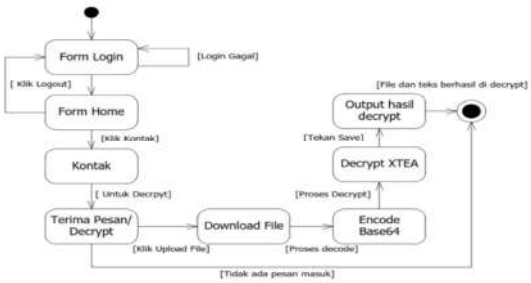
Pada keseluruhan program, prosesnya dapat dilihat pada *statechart* diagram program terlihat proses bagaimana cara aplikasi ini bekerja mulai dari tahap awal hingga akhir,dimana dari awal login lalu proses enkrip untuk pengiriman pesan dan proses dekrip untuk menerima pesan dimana akan dilakukan enkrip dan dekrip menggunakan algoritma XTEA dan *encode/decodeBase64* seperti pada gambar 4 :

Gambar 4 : Statechart diagram program

Proses enkripsi dan dekripsi dalam program dapat dilihat pada *statechart* diagram gambar 5.



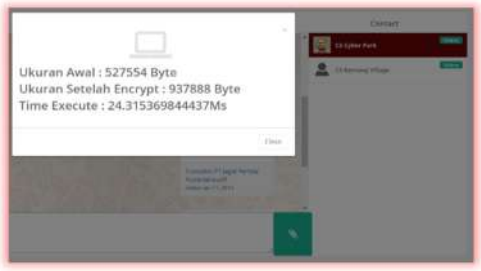
Gambar 5 : Statechart diagram program proses encrypt



Gambar 6 : Statechart diagram program proses decrypt

3.4 Hasil dan Pengujian Program

Pada bagian ini akan dijelaskan langkah-langkah dalam proses pengujian mulai dari tahap pengujian login ke dalam aplikasi yag telah dibuat, kemudian proses pengujian *encrypt* dan *decrypt*. Hasil dari pengujian yang dilakukan, file tersebut berhasil dilakukan proses pengamanan data, sehingga pihak lain tidak dapat melihat isi dari data yang sesungguhnya. Berikut merupakan prosesenkripsi arau pengiriman *file* yang dilakukan pada *filepdf*, pilih kontak tujuan pengiriman *file* lalu pilih pesan yang akan di *encrypt* dan kirim, sesuai pada gambar 7 :

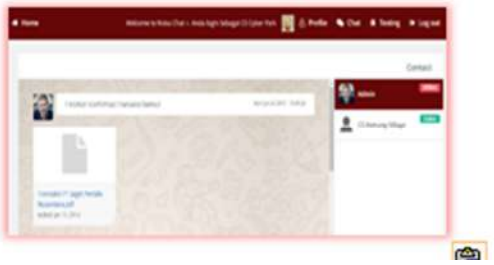


Gambar 7: Tampilan proses kirim file dan enkripsi berhasil

Dan berikut merupakan*filepdf*yang telah dilakukan proses pengamanan data, yang berisikan data acak :

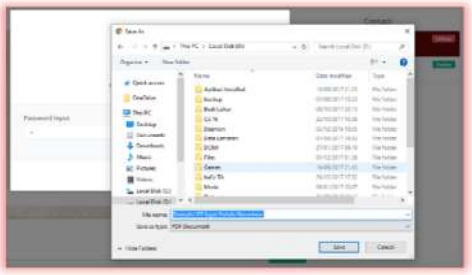
Gambar 8 : File hasil enkripsi

Berikut merupakan proses dekripsi atau menerima pesan berisikan *filepdf* yang sebelumnya dilakukan proses enkripsi :



Gambar 9 : Tampilan uji coba pesan masuk atau decrypt

lalu tekan *file* yang akan di *download* kemedian aplikasi akan mem proses *decrypt* dan meminta memasukan *passwordlogin* untuk men*downloadfile*, sampai akan muncul *browsefile* untuk menyimpan hasil *decrypt* seperti gambar 10.



Gambar 10 : Tampilan browse file

Tabel 1 merupakan tabel pengujian encrypt file ;

Tabel 1 : Tabel pengujian encrypt file < 2 MB

No	Nama File	Sebelum *Encrypt*		Ukuran file encrypt (bytes)	Lama Proses *Encrypt* (S)	Tambahan Ukuran File (%)
		Format File	Ukuran File (bytes)			
1	jadwal wb November 2017	xlsx	13804	24552	0,71	43,77 %
2	Resume	doc	39424	70092	2,19	43,75 %
3	Naik limit User ID	jpg	214723	381740	11,05	43,75 %
4	JPN	jpg	336371	598008	19,09	43,75 %
5	Logo UBL	png	358752	637784	21,80	43,75 %
6	Presentasi blog	pptx	545279	969388	30,12	43,75 %
7	Buka blokir ATM an. Andreas	jpg	976140	1735372	45,23	43,75 %`
8	Transaksi Pemeliharaan Data Sukardan	pdf	1325615	2356652	73,11	43,75 %
9	Maher Zain - Insha Allah (Arabic)	Mp3	1469518	2612480	79,36	43,75 %
10	Ecommerce	pptx	1561089	2775288	87,29	43,75 %
Rata-rata penambahan ukuran file setelah dilakukan *encrypt* dan rata-rata kecepatan proses *encrypt* per *byte* =						44%

Tabel 2 dan tabel 3 merupakan tabel pengujian *decrypt file* :

Tabel 2 : Tabel pengujian decrypt file < 2 MB

No	Nama File	Ukuran File (bytes)	Setelah *Decrypt*		Lama Proses *Decrypt* (MS)	Tambahan Ukuran File (%)
			Format File	Ukuran File (bytes)		
1	jadwal wb November 2017	24550	xlsx	13808	0,76	-43,77 %
2	Resume	70091	doc	39424	2,45	-43,75%
3	Naik limit User ID	381739	jpg	214728	14,29	-43,75%
4	JPN	598006	jpg	336376	22,77	-43,75%
5	Logo UBL	637782	png	358752	18,43	-43,75%
6	Presentasi blog	969387	pptx	545280	28,26	-43,75%
7	Buka blokir ATM an. Andreas	1735371	jpg	976144	50,66	-43,75%
8	Transaksi Pemeliharaan Data Sukardan	2356651	pdf	1325616	69,42	-43,75%
9	Maher Zain - Insha Allah (Arabic)	2612480	Mp3	1469520	76,53	-43,75%
10	Ecommerce	2775286	pptx	1561096	81,37	-43,75%
Rata-rata penambahan ukuran file setelah dilakukan *encrypt* dan rata-rata kecepatan proses *encrypt* per *byte* =						-44%

Tabel 3 : Tabel pengujian decrypt file > 2 M

No	Nama File	Ukuran File (bytes)	Setelah *Decrypt*		Lama Proses *Decrypt* (S)	Penambahan Ukuran File (%)
			Format File	Ukuran File (bytes)		
1	Flower	3760043	jpeg	2115024	123,75	-43,75%
2	UAS	4648064	doc	2614536	137,19	-43,75%
3	1211530231 (1)	5072758	pdf	2853424	178,22	-43,75%
4	Perancangan Program	5410432	pdf	3043368	191,51	-43,75%
5	Plain White	5562400	Mp3	3128848	165,48	-43,75%
Rata-rata penambahan ukuran file setelah dilakukan *decrypt* dan rata-rata kecepatan proses *decrypt* per *byte* =						**-43,75%**

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan serta uji coba dapat disimpulkan sebagai berikut :

a. Pengamanan pesan dapat diamankan dengan algoritma kriptografi XTEA dan *Base64.*
b. Data tidak dapat dibuka oleh pihak yang tidak memiliki *user* id.
c. Berdasarkan *file* yang pernah di coba lama eksekusi tergantung pada isi dan ukuran *file*, semakin banyak dan besar ukuran *file* akan semakin lama di eksekusi.
d. Ukuran *file* untuk dilampirkan yang paking optimal untuk aplikasi ini adalah ukuran dibawah 1MB.
e. Rata rata perubahan *file* untuk *encrypt* dan *decrypt* bertambah sekitar 44% dan berkurang 44%.

Pengembangan yang perlu dilakukan penelitian berikutnya adalah sebagai berikut :

a. Agar program dapat melakukan proses encrypt dan decrypt dengan waktu yang lebih singkat.
b. Agar dapat melakukan proses encrypt dan decrypt dengan ukuran yang lebih besar.
c. Jumlah dile lampiran yang disisipkan lebih dari satu file
d. Agar pengujian mendapatkan hasil yang lebih bauk,sebaiknya menggunkan spesifikasi hardware yang lebih tingg**i.**

5. DAFTAR PUSTAKA

[1] Al Meer H. Mohamed (2017) dalam jurnal dengan judul "Programmable SoC for an XTEA Encryption Algorthm Using A Co-Design Environment Replication Performance Approach", *ISSN 2327-5227, Issue 5 Volume 40-59.*

[2] Ariyus, Doni 2008, Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi.

[3] Ballal, Vignesh 2017, A Study and Comparison of Lightweight Cryptographic Algorithm, *ISSN :2278-8735.Volume 12, Issue 4, Ver.II (Jul.-Aug.2017)*.

[4] Isobe., Takanori, Shibutami., Kyoji 2012, Security Analysis of the Lightweight Block Ciphers XTEA, LED and Piccolo, *ACISP 2012, LNCS 7372, Springer-Verlag Berlin Heideberg 2012*.

[5] Mandalamaya 2014, Pengertian chatting dan Sejarah chatting, diakses 30 Oktober 2017, <http://www.mandalamaya.com/pengertian-chatting-dan-sejarah-chatting/>.

[6] Pramudya, Deva 2014, Algoritma Base64, diakses 20 Maret 2017, <http://pramudyasuryawan.blogspot.co.id/2014/11/algoritma-base64.html>

[7] Shweta Gaba, Iti Aggarwal and Dr. Sujata Pandey. "Design of Efficenit XTEA Using Verilog", *Int. J. of Scien. And Res. Public, vol. 2, no.6, Jun. 2012.*

[8] S. Sopna, R Muthaiah 2016, Implementasi of High Throughput Extended Tiny Encryption Algorthm Block Cipher in Field Programmable Gate Array, *ISSN:0974-5645 Volume 9(29) August 2016.*

[9] Rumagia, Juliana Lili Kethrina 2014, Perancangan dan Implementasi Aplikasi enkripsi-Dekripsi folder menggunakan Algoritma XTEA, dilihat pada 22 September 2017, < http://repository.uksw.edu/bitstream/123456789/8703/3/T1_672009253_Full%20text.pdf >.

[10] William Stallings, Cryptography and NetworkSecurity: *Principles and Practice, 5th ed. New Jersey :Prentice Hall, 2011.*