Jurnal Sistem Komputer dan Kecerdasan Buatan
Vol 3 No.1 Tahun 2019

ANALISIS DAN SIMULASI ALGORITMA TEA (TINY ENCRYPTION ALGORITHM) UNTUK ENKRIPSI DAN DEKRIPSI PESAN TEXT MENGGUNAKAN CRYPTOOL2

Muhamad Femy Mulya[1], Nofita Rismawati[2]

[1]Program Studi Sistem Informasi, Tanri Abeng University, Jakarta, Indonesia
femy.mulya@tau.ac.id
[2]Program Studi Teknik Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia
novi.9001@gmail.com

Diterima 20 Agustus 2019
Disetujui 23 September 2019

***Abstract*— The development of data and information technology and communication at this time is an important part of everyday human life. Along with the times, the human need for information is increasing. as the development of data and information technology that is increasingly advanced and developing, the internet can no longer guarantee to provide data that is safe enough for its users. Data is very important, so the accuracy of the data is needed in the decision making process. The importance of data values causes the availability of data security, so that data falls precisely and accurately on the right party. In this research a data security simulation will be designed using encryption and decryption methods using the TEA (*Tiny Encryption Algorithm*) algorithm. In the TEA (*Tiny Encryption Algorithm*) encryption process, this study uses 64 rounds, where each round consists of, addition, key & data substitution, and XOR. The purpose of this study is to analyze, design and implement TEA (*Tiny Encryption Algorithm*) algorithm in the form of encryption and decryption simulation of text messages (both text and text files), so that text messages (both text and text files) will be safer when sent via email from the sender to the recipient. The software that will be used to create a simulation of the TEA (*Tiny Encryption Algorithm*) algorithm uses Cryptool2, which is an Open Source program used to describe the concepts of cryptography and cryptanalysis. The test results prove that this simulation can secure text messages (both text writing or text files) with different character lengths and keys. The longer and larger the size of the text message (either text or text file) and the key, the longer it will take for the encryption process.**

Keyword: *Text Messages*, *TEA* (*Tiny Encryption Algorithm*), *Cryptool2*

I. PENDAHULUAN

Masalah keamanan dan kerahasiaan data merupakan hal yang sangat penting dalam suatu organisasi maupun pribadi. Apalagi jika data tersebut berada dalam suatu jaringan komputer yang terhubung/terkoneksi dengan jaringan lain (contoh: jaringan internet dan intranet). Hal tersebut tentu saja akan menimbulkan resiko, bilamana informasi yang sensitif dan berharga tersebut diakses oleh orang-orang yang tidak berhak dan tidak memiliki kepentingan. Jika hal tersebut sampai terjadi, kemungkinan besar akan sangat merugikan, bahkan membahayakan orang yang mengirim pesan maupun menerima pesan. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak tersebut akan memiliki kemungkinan rusak bahkan hilang yang akan menimbulkan kerugian material yang besar.

Oleh karena itu, untuk menghindari agar hal tersebut tidak terjadi, digunakanlah sebuah program khusus proteksi/enkripsi data. Saat ini banyak beredar program khusus proteksi data, pada umumnya program tersebut tidak hanya menyediakan satu metoda saja, tetapi beberapa jenis sehingga kita dapat memilih yang menurut kita paling aman. Dewasa ini, dalam dunia dengan arus informasi yang semakin global, kriptografi telah menjadi suatu bagian yang tidak dapat dipisahkan dari sistem keamanan jaringan. Ada berbagai algoritma kriptografi yang sekarang ini telah dan sedang dikembangkan, salah satunya diantaranya algoritma kunci simetris ataupun asimetris (pembagian berdasarkan kunci). Algoritma TEA merupakan algoritma penyandian *block cipher* yang menggunakan proses *feistel network* dengan panjang kunci 128 bit, dengan cara memproses 64-bit input sekali waktu dan menghasilkan 64-bit output.

Penelitian ini penting dilakukan, karena pada penelitian-penelitian yang sudah dilakukan hanya menggunakan Algoritma TEA *Tiny Encryption Algorithm*) dengan round sebanyak 16 ataupun 32 round, sedangkan pada penelitian ini menggunakan Algoritma TEA *Tiny Encryption Algorithm*) dengan round sebanyak 64. Selain itu, pada penelitian ini juga akan dilakukan uji coba terhadap simulasi enkripsi dan dekripsi pesan *text* (berupa *text writing* ataupun *text file*) Menggunakan Algoritma TEA (Tiny Encryption Algorithm) dengan *software* Cryptool2.

Dari uraian yang telah diberikan, pada penelitian ini akan dijawab permasalahan bagaimana pemanfaatan Algoritma TEA (*Tiny Encryption Algorithm*) untuk proses enkripsi dan dekripsi pesan *text* (berupa *text writing* ataupun *text file*) bisa diimplementasikan dalam bentuk simulasi proses enkripsi dan dekripsi pada suatu proses pengiriman pesan melalui email, sehingga akan menjaga integritas dan keamanan data serta informasi yang saat ini sudah semakin berkembang.

Tujuan dari penelitian ini adalah untuk menganalisa, merancang dan mengimplementasikan Algoritma TEA (*Tiny Encryption Algorithm*) dalam bentuk simulasi enkripsi dan dekripsi dari pesan *text* (berupa *text writing* ataupun *text file*), sehingga pesan *text* (berupa *text writing* ataupun *text file*) akan lebih aman pada saat dikirimkan melalui email dari *sender* ke *receiver*. Perangkat lunak yang digunakan untuk membuat simulasi Algoritma TEA (*Tiny Encryption Algorithm*) ini menggunakan Cryptool2.

II. LANDASAN TEORI

A. Enkripsi dan Dekripsi

Enkripsi adalah proses penggunaan algoritma yang kompleks untuk mengkonversi pesan (*plaintext* atau *cleartext*) ke suatu pesan terenkripsi (*chipertext*)[1]. Hal ini ditujukan untuk mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Enkripsi dapat digunakan untuk tujuan keamanan, tetapi teknik lain masih diperlukan untuk membuat komunikasi yang aman, terutama untuk memastikan integritas dan autentikasi dari sebuah pesan.

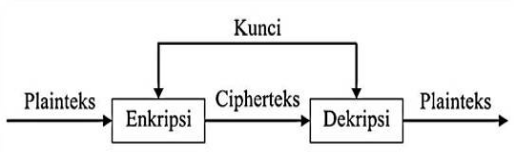
Deskripsi adalah proses penggunaan algoritma yang kompleks untuk mengkonversi pesan terenkripsi (*chipertext*) ke suatu pesan (*plaintext* atau *cleartext*). Bit Kunci dan Blok Data banyak sekali jenis enkripsi dan setiap enkripsi dibedakan berdasar besar bit kunci dan besar blok datanya. Dari mulai bit kunci 8-bit, sampai 256-bit. dan dari 64-bit hingga 512-bit besar blok data. Bit Kunci dan Blok Data penting karena kekuatan dari enkripsi terletak dalam tiga hal, yaitu besar Bit Kunci, besar Blok Data, dan metode pengulangan yang dilakukan didalamnya. Semakin besar Blok Kunci, maka akan semakin kecil Blok Data dan semakin banyak pengulangan yang dilakukan maka enkripsi tersebut bisa dibilang cukup tangguh, dan begitu juga sebaliknya.

B. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi [2]. Pesan yang akan dienkripsi disebut sebagai *plaintext* (teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah *plaintext* melibatkan penggunaan suatu bentuk kunci. Pesan *plaintext* yang telah dienkripsi (atau dikodekan) dikenal sebagai *ciphertext* (teks sandi) [3].

Algoritma kriptogarfi disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plaintext* dan himpunan yang berisi *ciphertext*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut.

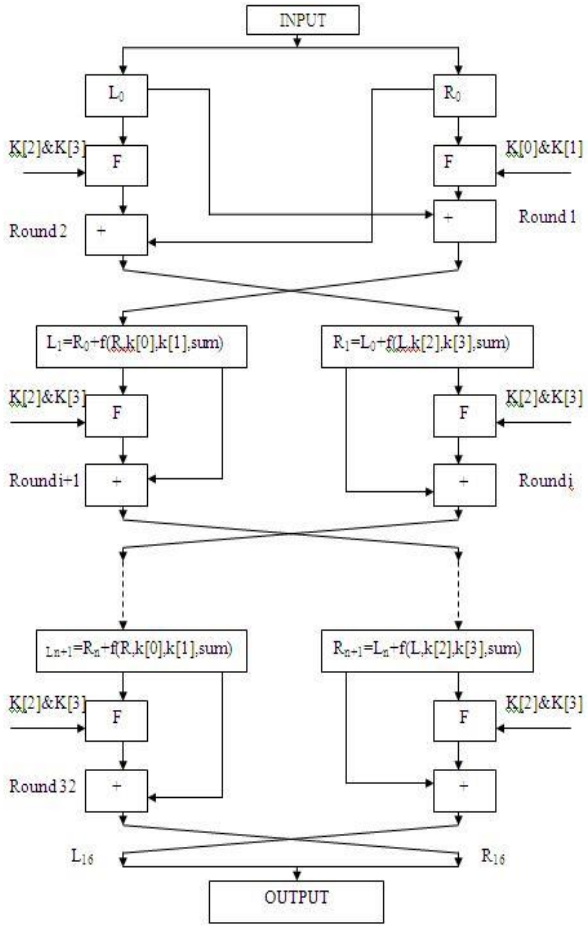
Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi [4]. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai skema diperlihatkan pada Gambar 1 [3].



Gambar 1. Skema Enkripsi dan Dekripsi dengan menggunakan Kunci [3]

C. Algoritma Tiny Enciption Algorithm (TEA)

Tiny Encription Algorithm (TEA) merupakan suatu algoritma sandi yang diciptakan oleh David Wheeler dan Roger Needham dari Computer Laboratory, Cambridge University, England pada bulan November 1994 [4]. Algoritma TEA merupakan algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional yaitu algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi.



Gambar 2. Alur Algoritma TEA[5]

Algoritma ini merupakan algoritma penyandian *block cipher* yang dirancang untuk penggunaan *memory* yang seminimal mungkin dengan kecepatan proses yang maksimal. Sistem penyandian TEA menggunakan proses *feistel network* dengan menambahkan fungsi matematik berupa penambahan dan pengurangan sebagai operator pembalik selain XOR [5]. Hal ini dimaksudkan untuk menciptakan sifat non-linearitas. Pergeseran dua arah (ke kiri dan ke kanan) menyebabkan semua bit kunci dan data bercampur secara berulang ulang, bisa dilihat pada gambar 2.

Proses diawali dengan *input-bit text* sebanyak 64-bit, kemudian 64-bit *text* tersebut dibagi menjadi dua bagian, yaitu sisi kiri (*L0*) sebanyak 32-bit dan sisi kanan (*R0*) sebanyak 32-bit. Setiap bagian *text* akan dioperasikan sendiri-sendiri. *R0* (*Z*) akan digeser kekiri sebanyak empat (4) kali dan ditambahkan dengan kunci k[0], sementara itu *Z* ditambah dengan sum (delta) yang merupakan konstanta.

Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Langkah selanjutnya di-XOR-kan dengan hasil penambahan antara *Z* yang digeser ke kanan sebanyak lima (5) kali dengan kunci *k[1].* Hasil tersebut kemudian ditambahkan dengan *L0 (Y)* yang akan menjadi *R1*. Sisi sebelah kiri akan mengalami proses yang sama dengan sisi sebelah kanan. *L0 (Y)* akan digeser ke kiri sebanyak empat (4) kali lalu ditambahkan dengan kunci *k[2],* sementara itu, *Y* ditambah dengan *sum* (delta). Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya.

Langkah selanjutnya di-XOR-kan dengan hasil penambahan antara *Y* yang digeser ke kanan sebanyak lima (5) kali dengan kunci k[3]. Hasil tersebut kemudian ditambahkan dengan *R0 (Z)* yang akan menjadi *L1*.

Adapun langkah-langkah penyandian dengan algoritma TEA dalam satu *cycle* (dua *round*) sebagai berikut:

1. Pergeseran *(shift)*
   Blok *text* terang pada kedua sisi yang masing masing sebanyak 32-bit akan digeser kekiri sebanyak empat (4) kali dan digeser ke kanan sebanyak lima (5) kali.
2. Penambahan
   Langkah selanjutnya setelah digeser kekiri dan kekanan, maka *Y* dan *Z* yang telah digeser akan ditambahkan dengan kunci k[0]-k[3]. Sedangkan *Y* dan *Z* awal akan ditambahkan dengan *sum* (delta).
3. Peng-XOR-an
   Proses selanjutnya setelah dioperasikan dengan penambahan pada masing-masing register maka akan dilakukan peng-XOR-an

dengan rumus untuk satu *round* adalah sebagai berikut:

- Hasil penyandian dalam satu *cycle* satu blok *text* terang 64-bit menjadi 64-bit teks sandi adalah dengan menggabungkan *Y* dan *Z*.
- Untuk penyandian pada *cycle* berikutnya *Y* dan *Z* ditukar posisinya, sehingga *Y1* menjadi *Z1* dan *Z1* menjadi *Y1* lalu dilanjutkan proses seperti langkah-langkah diatas sampai dengan *16 cycle (32 round).*

4. *Key Schedule*

   Algoritma TEA menggunakan *key schedule*-nya sangat sederhana. Yaitu kunci k[0] dan k[1] konstan digunakan untuk *round* ganjil sedangkan kunci k[2] dan k[3] konstan digunakan untuk *round* genap.

5. Dekripsi

   Proses dekripsi sama halnya seperti pada proses penyandian yang berbasis *feistel cipher* lainnya. Yaitu pada prinsipnya adalah sama pada saat proses enkripsi. Hal yang berbeda adalah penggunaan *text* sandi sebagai input dan kunci yang digunakan urutannya dibalik. Proses dekripsi semua *round* ganjil menggunakan k[1] terlebih dahulu kemudian k[0], demikian juga dengan semua *round* genap digunakan k[3] terlebih dahulu kemudian k[2]. Rumus untuk enkripsi ditunjukkan oleh Persamaan (1) dan (2).

$$L_1 = L_0 + f\ (R_0, k[0], k[1], sum) \qquad (1)$$

$$R_1 = R_0 + f\ (L_0, k[2], k[3], sum) \qquad (2)$$

Ket:
L_1 = *round* satu pada *round* ganjil
L_0 = *round* nol pada *round* ganjil
f = fungsi
R_1 = *round* satu pada *round* genap
R_0 = *round* nol pada *round* genap
K = kunci

Rumus proses dekripsi ditunjukkan oleh Persamaan (3) dan (4).

$$L_1 = L_0 + f\ (R_0, k[1], k[0], sum) \qquad (3)$$

$$R_1 = R_0 + f\ (L_0, k[3], k[2], sum) \qquad (4)$$

Ket:
L_1 = *round* satu pada *round* ganjil
L_0 = *round* nol pada *round* ganjil
f = fungsi
R_1 = *round* satu pada *round* genap
R_0 = *round* nol pada *round* genap
K = kunci

D. Cryptool2

CrypTool2 merupakan sebuah perangkat lunak yang digunakan untuk menggambarkan atau mendeskripsikan konsep kriptografi dan kriptoanalisis. Secara umum aplikasi Cryptool2 ini mendukung dua algoritma kriptografi, yaitu algoritma kriptografi *modern* dan algoritma kriptografi *classic*. CrypTool2 adalah sebuah program gratis (*Open Source*) yang dikembangkan oleh *University of Kassel* (Jerman) [6].

Perangkat lunak (*software*) ini sangat bermanfaat bagi orang yang sedang mempelajari atau belajar mengenai kriptografi. Akan tetapi bagi pengguna umum maupun praktisi komputer, *software* ini amat sangat bermanfaat sekali terutama bagi yang ingin memperdalam pengetahuan tentang algoritma kriptografi untuk proses enkripsi dan dekripsi. Adapun contoh algoritma yang dapat diselesaikan dengan *software* ini antara lain seperti: Caesar, Vigenere, MD5, TEA, AES, DES, RSA serta lebih dari 300 algoritma lainnya. CrypTool2 juga menyediakan antarmuka (GUI) pengguna grafis untuk pemrograman visual [7].

Kelebihan Cryptool2 seperti: Memberikan antarmuka (*GUI*) untuk berbagai algoritma kriptografi, termasuk visualisasi yang bisa diatur untuk setiap parameternya, kemudian tampilan *GUI software* menggunakan antarmuka seperti Office 2007.

III. METODE PENELITIAN

Untuk menganalisa dan merancang simulasi enkripsi dan dekripsi pada pesan *text* (berupa *text writing* ataupun *text file*) dengan menggunakan Algoritma TEA (*Tiny Encryption Algorithm*) pada penelitian ini, akan digunakan adalah penelitian dengan studi literatur dan metode kuantitatif. Pada studi literatur dilakukan studi pustaka yang membahas teknik enkripsi dan dekripsi dengan Algoritma TEA (*Tiny Encryption Algorithm*).

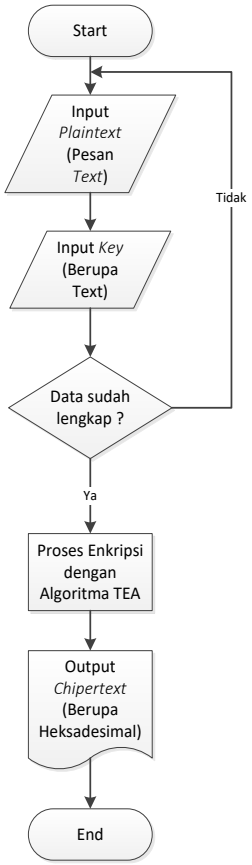
Kemudian pada metode kuantitatif dilakukan menggunakan metode penelitian eksperimental, yaitu dengan melakukan eksperimen terhadap variabel-variabel *input* untuk menganalisis *output* yang dihasilkan. Penelitian Eksperimental

merupakan bentuk penelitian dimana peneliti (eksperimenter) dengan sengaja melakukan uji coba terhadap objek yang terdapat pada perangkat lunak (*software*) simulasi, selanjutnya dilakukan pengamatan dan pencatatan hasil uji coba yang dilakukan, kemudian melihat hubungan diberikan dan reaksi yang muncul dari Proses. Adapun dua proses yang akan dilakukan untuk simulasi pada algoritma ini, yaitu proses enkripsi dan dekripsi. Enkripsi adalah proses pengubahan *plaintext* (pesan *text*) kedalam *chipertext* (heksadesimal) dan dekripsi adalah proses pengubahan *chipertext* (heksadesimal) kedalam *plaintext* (pesan *text*).

IV. HASIL DAN PEMBAHASAN

A. *Analisis Alur Kerja Simulasi Enkripsi dan Dekripsi menggunakan Algoritma TEA (Tiny Encryption Algorithm)*

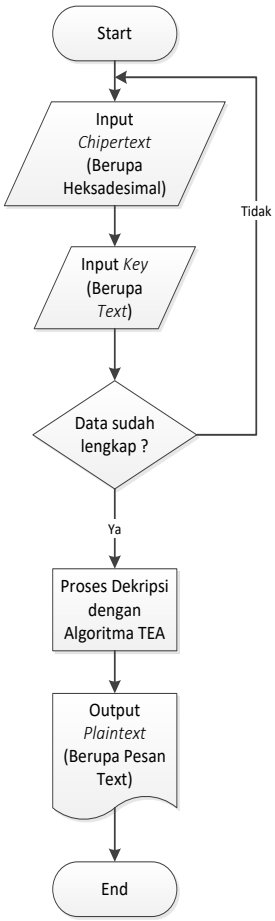
Analisis ini dilakukan guna memahami alur kerja simulasi Algoritma TEA (*Tiny Encryption Algorithm*) untuk proses enkripsi dan dekripsi pada pesan *text*. Berikut adalah *flowchart* untuk simulasi enkripsi untuk pesan *text* dengan Algoritma TEA (*Tiny Encryption Algorithm*).



Gambar 3. Flowchart simulasi enkripsi pada Algoritma TEA (*Tiny Encryption Algorithm*)

Pada *flowchart* simulasi enkripsi Algoritma TEA (*Tiny Encryption Algorithm*) terlihat pada gambar 3, langkah pertama untuk melakukan proses enkripsi pada algoritma TEA, dibutuhkan inputan berupa *plaintext* (berupa pesan *text)*, serta inputan *key/kunci* (ini harus diingat, karena akan dipakai untuk proses dekripsi algoritma TEA ke pesan *text*). Kemudian jika semua data telah selesai diinput, proses selanjutnya akan dilakukan perhitungan/proses enkripsi pada algoritma TEA, lalu akan dihasilkan *output* berupa *chipertext* (format heksadesimal).

Berikut adalah *flowchart* untuk simulasi dekripsi pada algoritma TEA, yang outputnya berupa *plaintext* (pesan *text*).



Gambar 4. Flowchart simulasi dekripsi pada Algoritma TEA (*Tiny Encryption Algorithm*)

Pada *flowchart* simulasi dekripsi pada algoritma TEA terlihat pada gambar 4, menjelaskan bagaimana proses dekripsi *chipertext* (dalam format heksadesimal) supaya bisa menampilan isi *planitext* (berupa pesan *text*). Langkah pertama masukkan *chipertext* (format heksadesimal) hasil output dari enkripsi pada algoritma TEA, kemudian masukkan pula *input key* yang dimasukkan sama persis saat

proses enkripsi pada algoritma TEA. Selanjutnya jika semua data telah selesai diinput, maka proses berikutnya akan dilakukan perhitungan dekripsi dengan algoritma TEA, dengan demikian akan dihasilkan output berupa pesan *text* (*plaintext*).

B. *Simulasi Enkripsi Algoritma TEA (Tiny Encryption Algorithm) dengan Cryptool2*

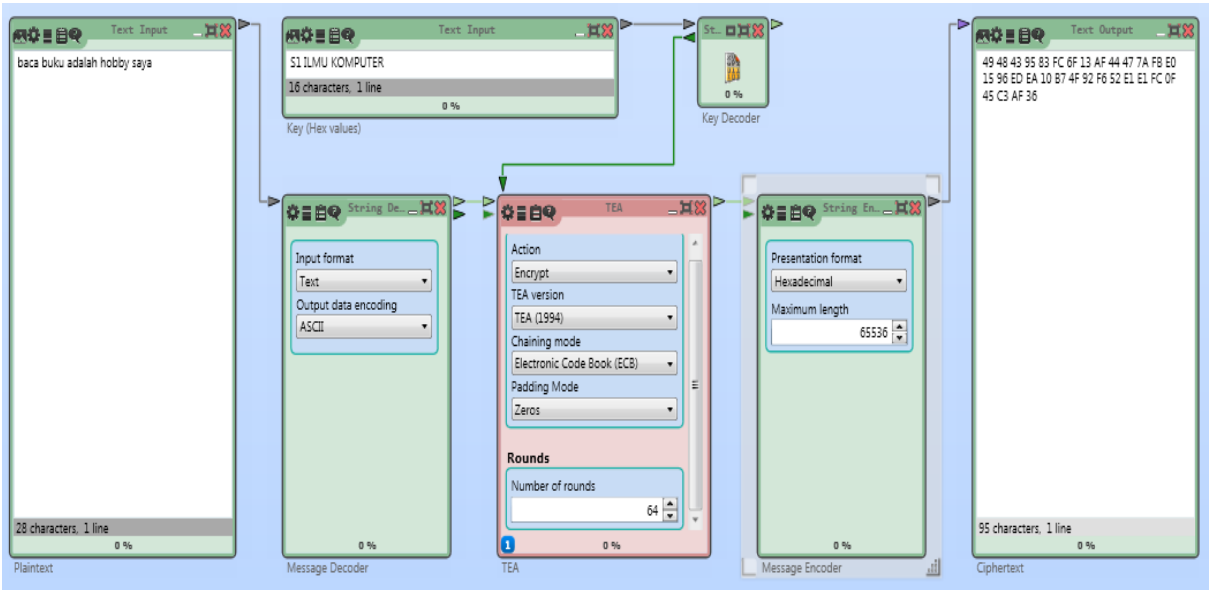
Untuk membuat simulasi enkripsi pada algoritma TEA berbasis cryptool2 kita harus membuat desain seperti pada gambar 5 (desain mengikuti alur *flowchart* yang sudah dibuat). Adapun beberapa *Properties*/*Tools* yang dibutuhkan antara lain sebagai berikut:

1. *Text input* sebanyak 2 (dua) buah, yang satu digunakan untuk pesan text yang akan dirubah ke *chipertext*. Kemudian yang kedua digunakan untuk *input key* (kunci pada algoritma TEA).
2. Satu buah *message decoder* yang akan digunakan untuk merubah input format *text* menjadi format ASCII.
3. Satu buah *Key decoder* yang akan digunakan untuk merubah input format *text* menjadi format UTF-8.
4. Satu buah algoritma TEA, kemudian kita pilih actionnya untuk Encrypt (enkripsi) dan *Number of Rounds* 64.
5. Satu buah *message encoder* yang akan digunakan untuk merubah hasil proses perhitungan algoritma TEA ke format heksadesimal.
6. Satu buah *text output* yang akan digunakan untuk menampilan *chipertext* (dalam format heksadesimal).

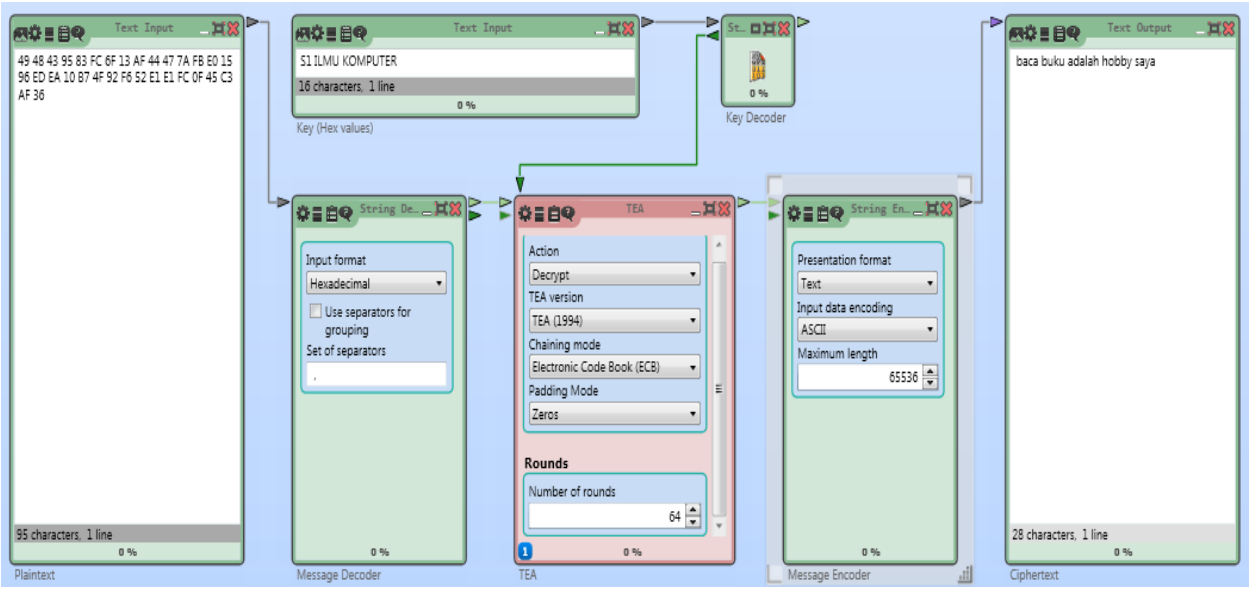
C. *Simulasi Dekripsi Algoritma TEA (Tiny Encryption Algorithm) dengan Cryptool2*

Untuk membuat simulasi dekripsi pada algoritma TEA berbasis cryptool2 kita harus membuat desain seperti pada gambar 6 (desain mengikuti alur *flowchart* yang sudah dibuat). Adapun beberapa *Properties*/*Tools* yang dibutuhkan antara lain sebagai berikut:

1. *Text input* sebanyak 2 (dua) buah, yang satu digunakan untuk *chipertext* yang akan dirubah ke *plaintext* (pesan *text*). Kemudian yang kedua digunakan untuk *input key* (kunci yang dimasukkan harus sama seperti proses enkripsi).
2. Satu buah *message decoder* yang akan digunakan untuk merubah input format heksadesimal menjadi format ASCII.
3. Satu buah *Key decoder* yang akan digunakan untuk merubah input format *text* menjadi format UTF-8.
4. Satu buah algoritma TEA, kemudian kita pilih actionnya untuk Decrypt (dekripsi) dan *Number of Rounds* 64 (*round* harus sama seperti saat proses enkripsi).
5. Satu buah *message encoder* yang akan digunakan untuk merubah hasil proses perhitungan algoritma TEA ke format pesan *text*.
6. Satu buah *text output* yang akan digunakan untuk menampilan *plaintext* (dalam format *text*).



Gambar 5. Simulasi Enkripsi pada Algoritma TEA (*Tiny Encryption Algorithm*) dengan Cryptool2



Gambar 6. Simulasi Dekripsi pada Algoritma TEA (*Tiny Encryption Algorithm*) dengan Cryptool2

D. Uji Coba Enkripsi dan Dekripsi Pesan Text Menggunakan Algoritma TEA (Tiny Encryption Algorithm) dengan Cryptool2

Pada penelitian ini akan dilakukan pengujian terhadap simulasi enkripsi dan dekripsi pesan *text* dengan algoritma TEA menggunakan perangkat lunak cryptool2. Kemudian akan dilakukan uji coba sampling sebanyak 10 kali uji coba untuk masing-masing proses enkripsi dan dekripsi terhadap pesan *text* dengan algoritma TEA, lalu untuk sampling yang digunakan adalah ukuran pesan *text* (dengan format *.txt), serta variabel pembanding berupa waktu proses untuk enkripsi dan dekripsi. Adapun hasil uji cobanya terlihat pada tabel 1 sebagai berikut.

Tabel 1. Hasil Uji Coba Enkripsi dan Dekripsi Pesan *Text* Menggunakan Algoritma TEA (*Tiny Encryption Algorithm*) dengan Cryptool2

No	Pesan Text Asli		Enkripsi		Dekripsi	
	Nama File (*.txt)	Ukuran Pesan Text original (Byte)	Ukuran (Byte)	Waktu (ms)	Ukuran (Byte)	Waktu (ms)
1	Sampel 1	12,56	10,44	0,13	12,56	0,11
2	Sampel 2	54,67	45,48	0,51	54,67	0,47
3	Sampel 3	107,54	86,10	0,92	107,54	0,77
4	Sampel 4	534,89	494,11	1,67	534,89	1,55
5	Sampel 5	1132,11	1036,13	2,78	1132,11	2,19
6	Sampel 6	5651,23	5121,69	4,82	5651,23	3,84
7	Sampel 7	13212,45	11627,78	9,73	13212,45	8,70
8	Sampel 8	57652,21	51728,33	14,34	57652,21	12,23
9	Sampel 9	114232,78	93294,29	27,66	114232,78	23,88
10	Sampel 10	521342,66	446294,30	39,33	521342,66	32,98

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian dan hasil pengujian yang dilakukan terhadap simulasi enkripsi dan dekripsi pada pesan *text* menggunakan algoritma TEA (*Tiny Encryption Algorithm*), maka didapatkan kesimpulan sebagai berikut:

1. Ukuran pesan *text* (*byte*) sangat mempengaruhi lamanya proses enkripsi dan dekripsi pesan. Semakin besar ukuran pesan *text (byte)*, maka akan semakin lama waktu (ms) proses yang diperlukan perangkat lunak Cryptool2 untuk melakukan enkripsi pesan *text*.
2. Ukuran *text file (byte)* dari hasil proses enkripsi cenderung naik ukuran *text file (byte)* dari *file* originalnya. Sedangkan Ukuran *text file (byte)* dari hasil proses dekripsi sama dengan ukuran *text file (byte)* dari *file* originalnya.
3. Estimasi waktu (ms) pada proses enkripsi dan dekripsi pesan *text* menggunakan algoritma TEA, yang tercepat dalam proses nya adalah dekripsi pesan *text*.

B. Saran

Berdasarkan hasil dari penelitian yang telah dilakukan oleh peneliti, berikut ini beberapa saran untuk pengembangan simulasi dengan perangkat lunak Cryptool2, antara lain:

1. Dalam melakukan proses enkripsi, sebaiknya usahakan untuk tidak menggunakan kunci yang sama untuk mengenkripsi pesan *text* yang berbeda. Selanjutnya, agar kinerja algoritma lebih optimal sebaikya algoritma TEA digabungkan dengan algoritma kriptografi asimetris (*hybrid cryptography*), dimana algoritma TEA hanya digunakan untuk proses enkripsi dan deskripsi pesan text, sementara untuk pembentukan kunci digunakan algoritma asimetris agar kunci yang didistribusikan tetap aman.
2. Perangkat lunak Cryptool2 diharapkan pada proses simulasi enkripsi dan dekripsi untuk algoritma TEA dapat men-*support* file ekstensi selain *.txt seperti file bertipe *.pdf, *.jpeg, *.doc, video, dan audio.

DAFTAR PUSTAKA

[1] N. Nurdin, "IMPLEMENTASI ALGORITMA TEA DAN FUNGSI HASH MD4 UNTU ENKRIPSI DAN DEKRIPSI DATA," *TECHSI - J. Tek. Inform.*, vol. 5, no. 1, Apr. 2013.

[2] W. Stallings *et al.*, *CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE SEVENTH EDITION GLOBAL EDITION*. 2017.

[3] F. Nandar Pabokory, I. Fitri Astuti, and A. Harsa Kridalaksana, "IMPLEMENTASI KRIPTOGRAFI PENGAMANAN DATA PADA PESAN TEKS, ISI FILE DOKUMEN, DAN FILE DOKUMEN MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD," 2015.

[4] A. Zelvina, S. Effendi, and D. Arisandi, "Perancangan Aplikasi Pembelajaran Kriptografi Kunci Publik ElGamal Untuk Mahasiswa," *Dunia Teknol. Inf. - J. Online*, vol. 1, no. 1, Dec. 2012.

[5] L. Liana, L. Liana, S. Sutardi, and N. F. Muchlis, "APLIKASI ENKRIPSI DAN DEKRIPSI DATA MENGGUNAKAN TINY ENCRYPTION ALGORITHM (TEA) BERBASIS JAVA," *semanTIK*, vol. 4, no. 1, pp. 39–48, Jul. 2018.

[6] "CodeSaya | Yuk belajar Kriptografi atau Enkripsi lebih mudah dengan software "CrypTool"" [Online]. Available: https://codesaya.com/a/yuk-belajar-kriptografi-atau-enkripsi-lebih-mudah-dengan-software-cryptool-rhdbdvvomb/. [Accessed: 04-Sep-2019].

[7] N. Rismawati and M. F. Mulya, "Analisis dan Perancangan Simulasi Enkripsi dan Dekripsi pada Algoritma Steganografi untuk Penyisipan Pesan Text pada Image menggunakan Metode Least Significant Bit (LSB) Berbasis Cryptool2," *Fakt. Exacta*, vol. 12, no. 2, pp. 132–144, Jul. 2019.