Implementasi Enkripsi dan Dekripsi File Dokumen menggunakan Metode Modifikasi Algoritma *Tiny Encryption Algorithm*

Ellsa Wahyu Candra Pujiarwoko[1], Aditya Prapanca[2]

[1,2]Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[1]ellsapujiarwoko16051204005@mhs.unesa.ac.id
[2]adityaprapanca@unesa.ac.id

***Abstrak*— Keamanan data menjadi hal yang paling dibutuhkan bagi semua pengguna komunikasi modern. Pada era digitalisasi ini terdapat banyak celah yang dapat digunakan untuk melanggar kebijakan terkait keamanan data. Untuk itu, perkembangan teknik pengamanan data juga telah dilakukan untuk mengatasi celah tersebut. Banyak penelitian telah dilakukan untuk menciptakan teknik pengamanan data untuk dapat mencegah adanya pelanggaran hak terhadap data yang dikirimkan melalui sistem komunikasi modern. Kriptografi merupakan salah satu teknik pengamanan data yang sering digunakan karena dapat diterapkan pada semua tipe data. Dalam penerapan kriptografi dibutuhkan teknik tertentu untuk melakukan rekonstruksi pada data asli. Tiny Encryption Algorithm merupakan salah satu algoritma yang digunakan dalam enkripsi data. Algortima TEA bekerja dengan konsep simetris enkripsi dengan kecepatan pemrosesan yang maksimal namun dengan konsumsi memori yang sedikit. Algoritma TEA bekerja menggunakan struktur feissel network yang membagi bit-input menjadi dua bagian sama besar dalam 64 round. Pada penelitian ini dilakukan observasi terhadap hasil modifikasi algortima TEA jumlah bit-input yang diproses dan jumlah round yang dilakukan dalam proses enkripsi dan dekripsi. Modifikasi dilakukan untuk mengetahui pengaruh jumlah bit input dan round terhadap waktu pemrosesan yang dibutuhkan dan ukuran dokumen hasil dari enkripsi maupun dekripsi. Hasilnya, modifikasi pada jumlah bit-input maupun putaran yang terjadi pada proses enkripsi maupun dekripsi dokumen berpengaruh pada hasil pemrosesan, yaitu dari sisi waktu dan ukuran dokumen hasil pemrosesan.**

***Kata Kunci* — Keamanan Data, Kriptografi, Enkripsi, PDF, Modifikasi TEA**

I. PENDAHULUAN

Perkembangan ilmu pengetahuan membawa dampak signifikan terhadap keberadaan teknologi di kehidupan manusia saat ini. Teknologi hampir menyentuh seluruh lapisan masyarakat melalui berbagai sektor pekerjaan. Mulanya teknologi tercipta untuk meringankan beban perhitungan sederhana dan terus berkembang menjadi mesin canggih yang telah dibekali berbagai kemampuan dan kecerdasan yang serupa dengan manusia. Dibandingkan dengan awal penciptaannya, teknologi saat ini telah mampu menyimpan dan mengolah data dalam jumlah yang tidak dapat didefinisikan.

Peran teknologi sedikit demi sedikit telah melampaui batas kemampuan manusia biasa. Dari sekian banyak keunggulan, terdapat celah kekhawatiran atas eksploitasi teknologi dalam bidang pekerjaan manusia. Penyimpanan dan pengolahan data melalui sebuah teknologi dinilai rentan terhadap pelanggaran akses oleh oknum yang tidak bertanggung jawab. Bagi penyedia jasa dan produk yang melibatkan transaksi melalui media digital maupun sektor pekerjaan lainnya, jaminan keamanan data merupakan prioritas utama yang ditawarkan.

Telah banyak dilakukan penelitian terkait pengamanan data untuk mendukung komunikasi di era digital ini. Bentuk-bentuk pengamanan data yang digunakan juga terus mengalami perkembangan dari waktu-waktu. Pengembangan yang dilakukan bertujuan agar teknik pengamanan data dapat dilakukan untuk semua jenis dokumen baik audio, visual, maupun audio visual.

Teknik pengamanan data bertujuan untuk melindungi data yang dikirimkan dari akses pengguna yang tidak sah, seperti menyalin, menyebar luaskan dan bahkan mengubah data yang asli. Dalam teknik keamanan data dikenal istilah kriptografi. Kriptografi merupakan teknik rekonstruksi data menjadi hal yang tidak bermakna dan tidak dapat dibaca menggunakan aturan-aturan tertentu. Pengirim akan mengirimkan data yang telah dienkripsi dan teknik mengembalikan data agar dapat dimengerti kembali, atau disebut dekripsi, hanya kepada penerima yang sah. Konsep ini yang dijadikan acuan dalam teknik kriptografi, yaitu data yang dikirimkan hanya dapat dimengerti oleh pengirim dan penerima yang sah saja.

Aspek keamanan informasi seperti, *data confidentiality, data integrity, authentication,* dan *non-repudiation* menjadi fokus utama dalam ilmu kriptografi modern. Kriptografi modern melibatkan disiplin ilmu lain seperti ilmu matematika, ilmu komputer, teknik elektro, ilmu komunikasi dan ilmu fisika. Algoritma kriptografi dirancang berdasarkan *computational hardness assumptions*, membuat algoritma tersebut sulit namun bukan tidak mungkin untuk dipecahkan secara teoritikal. Algoritma-algoritma dalam kriptografi terus dikembangkan dengan melibatkan teori-teori algoritma faktorisasi integer dan komputasi yang lebih cepat dan terus dievaluasi.

Kriptografi banyak digunakan dalam perdagangan elektronik, metode pembayaran berbasis *chip*, mata uang digital, *password* komputer dan komunikasi militer. Kriptografi dapat digunakan untuk mengamankan data dalam bentuk text, audio, visual, dan video. Penelitian [1] membandingkan dan mengevaluasi kinerja algoritma enkripsi dan dekripsi secara *real-time* pada signal audio. Algortima RSA dibandingkan dengan algoritma baru yang dirancangan berdasarkan kriptografi simetris. Pengujian ini menggunakan *software Matlab Simulink* untuk penerapan dan simulasi kedua algoritma yang diuji. Dari hasil perbandingan, signal audio yang telah dienkripsi dan didekripsi dengan signal audio asli

didapatkan hasil bahwa algoritma RSA menghasilkan signal audio dengan kualitas rendah, sedangkan algoritma yang diajukan menghasilkan signal audio yang sama dengan aslinya. Nilai error dari kedua algoritma juga diperhitungkan. Algortima RSA menunjukkan nilai *error* lebih tinggi daripada algoritma yang diajukan.

Dalam kriptografi dikenal istilah *ciphertext. Ciphertext* adalah hasil enkripsi dari *text* biasa menggunakan algoritma yang disebut *cipher. Ciphertext* berisikan informasi yang dijadikan baris-baris tulisan tak terbaca secara langsung oleh manusia atau komputer tanpa sandi yang tepat untuk mendekripsikan. Hal ini bertujuan untuk mencegah kehilangan informasi penting melalui peretasan. Pada beberapa kasus, pengirim sengaja mengembangkan algortima baru dengan skema yang hanya diketahui olehnya. Penelitian [2] mengembangkan algortima yang diberi nama Algoritma Nur. Skema yang dibangun pada penelitian ini menerapkan teknik kriptografi modern berbasis simetris kriptografi. Keamanan enkripsi pada algortima ini hanya bergantung pada kunci simetris yang digunakan bukan pada seberapa dikenal algoritma ini dimasyarakat. Untuk menambahkan kompleksitas pada algoritma ini digunakan mekanisme perkalian. Sistem yang dibangun diberi nama *Nur Aminuddin's Encryptor* dan terdapat dua teknik *data-reading,* enkripsi dan dekripsi. Beberapa cara untuk meningkatkan kualitas keamanan dapat dilakukan dengan menambahkan tingkat keamanan atau pengacakan. Untuk lebih meningkatkan keamanan, file yang diinginkan dapat dimasukkan kembali kedalam *Nur Aminuddin's Encryptor* untuk dienkripsi lagi. Memutar urutan data membuat data awal susah untuk ditemukan sehingga proses dekripsi hanya dapat diketahui oleh yang berhak. Selain itu, membagi file menjadi beberapa bagian sehingga ketika terjadi kebocoran data maka pihak yang tidak sengaja mendapatkan data tersebut tidak mendapatkan informasi seutuhnya.

Salah satu algoritma yang terkenal dikalangan peneliti adalah algoritma *Tiny Encryption Algorithm* (TEA). Algoritma ini popular karena sengaja dirancang untuk meminimalkan penggunaan memory namun kecepatan pemrosesan yang maksimal. Peneliti pada [3] telah merancang sebuah aplikasi yang memungkinkan dua orang saling bertukar pesan secara rahasia dengan menerapkan algoritma TEA untuk mengamankan pesan yang dikirimkan. Ukuran kunci yang digunakan dalam proses enkripsi dan dekripsi adalah 16 karakter atau lebih. Sistem membutuhkan waktu proses rata-rata 6.66 ms untuk mengenkripsi pesan dengan ukuran 11 *byte,* 5 *ms* untuk pesan berukuran 8 *byte*, 4 ms untuk enkripsi pesan 5 *byte* dan 2 ms untuk pesan dengan ukuran 3 *byte.* Semua pesan yang diujikan menggunakan kunci yang sama untuk proses enkripsi. Sedangkan untuk proses dekripsi, sistem membutuhkan waktu rata-rata 6.33 ms untuk pesan dengan ukuran 11 *byte,* 5 *ms* untuk pesan berukuran 8 *byte*, 2.33 ms untuk dekripsi pesan 5 *byte* dan 1.33 ms untuk pesan dengan ukuran 3 *byte.* Melalui pengujian ini dapat disimpulkan bahwa panjang pesan yang akan dienkripsi maupun didekripsi akan berpengaruh pada waktu pemrosesan yang dibutuhkan.

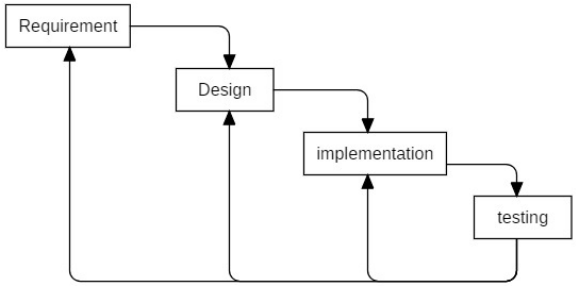
Selain itu, penelitian [4] berhasil menciptakan sebuah aplikasi yang mampu melakukan enkripsi dan dekripsi terhadap file berekstensi *txt* dan *docx* menggunakan kunci sepanjang 16 *bit.* Aplikasi yang dibangun diuji berdasarkan pemrosesan dan perubahan ukuran file yang telah diproses. Menurut hasil uji waktu pemrosesan, dibutuhkan waktu 15 ms untuk memroses pesan txt berukuran 95 *byte,* 31 *ms* untuk file txt berukuran 453 *byte*, 88 ms untuk file berukuran 1,682 *byte,* 130 *ms* untuk file berukuran 2,470 *byte*, dan 150 *ms* untuk file berukuran 3,642 *byte.* Sedangkan dari hasil uji ukuran file yang telah diproses, terdapat perubahan ukuran jika dibandingkan dengan ukuran file asli.

Pengamanan file PDF menggunakan algoritma TEA [5] dilakukan dengan menggunakan prosedur pengamanan sebagai berikut: *plaintext* dibagi menjadi 2 blok yang masing-masing berukuran 32 bit. *Key* yang digunakan berukuran 128 bit dibagi menjadi 4 blok dengan tiap blok berisi 32 bit. Proses enkripsi dan dekripsi dilakukan dalam *one cycle* sebanyak 32 kali dengan dua *round.* Sehingga total *round* pada proses dekripsi dan enkripsi adalah 64 *round.*

Melakukan moodifikasi jumlah bit input dan perulangan yang dilakukan dalam proses enkripsi menggunakan algoritma TEA yang ada akan lebih meningkatkan keamanan data, 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 [6]. Pada penelitian ini akan dibangun sebuah sistem yang mampu melakukan enkripsi dan dekripsi melalui implementasi modifikasi algoritma TEA pada jumlah bit-input atau perulangan untuk mengamankan dokumen yang berformat .txt .docx .pdf. Hasil modifikasi algoritma TEA akan dibandingkan dengan algoritma TEA asli dari segi waktu yang dibutuhkan untuk melakukan pemrosesan dan perubahan ukuran file yang telah diproses.

II. METODOLOGI PENELITIAN

Sistem dibangun menggunakan *software netbeans* dengan bahasa *Java*. Sistem yang dirancang dapat merekonstruksi file *PDF* untuk tujuan keamanan data dan mengembalikan ke bentuk semula agar dapat dimengerti oleh penerima. Proses pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1 Alur proses penelitian

Tahapan proses pada penelitian ini meliputi:

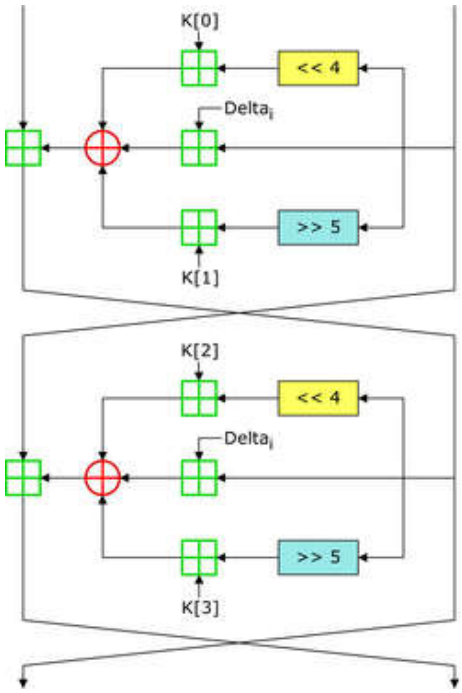
1. *Requirement*
   Pada tahap requirement dilakukan penjabaran kebutuhan dan fungsional sistem yang akan dibangun.
2. *Design*
   Dilakukan perancangan arsitektur sistem yang akan dibangun dan menghasilkan alur-alur proses yang akan dibuat pada sistem.
3. *Implementation*
   Tahap implementation merupakan proses perancangan sistem dengan mengubah skenario-skenario yang telah dibuat sebelumnya menjadi baris-baris kode.
4. *Testing*
   Setelah sistem berhasil dirancang akan dilakukan uji coba untuk mengetahui apakah sistem dapat menjalankan semua fungsional yang diharapkan.

Dalam ilmu kriptografi, algoritma TEA merupakan salah satu algoritma yang banyak digunakan dalam penyandian *block cipher*. Algoritma ini dirancang agar dapat menggunakan *memory* seminal mungkin namun dengan kecepatan komputasional yang maksimal [4]. Algoritma TEA beroperasi dengan *feistel network* yang memproses 64-bit input dengan memecahnya menjadi dua blok dengan ukuran sama dan menggabungkan kembali untuk menghasilkan 64-bit *output*.

A. *Enkripsi Algoritma TEA*

Proses enkripsi diawali dengan memecah *bit-input* menjadi dua blok masing-masing berisi 32-bit yang disimpan dalam v_0 dan v_1 dan 128-bit kunci yang dipecah kedalam empat variable, K[0], K[1], K[2], K[3], K[4] dengan panjang 32-bit. Konstanta delta (Delta [i]) yang biasa digunakan pada algoritma ini adalah 2654435769 dan 0x9E3779B9 [7]. Pergeseran dua arah (kiri dan kanan) membuat semua bit kunci dan data tercampur secara berulang-ulang. Algoritma ini beroperasi dengan tambahan fungsi aritmetika berupa penjumlahan dan pengurangan sebagai operator pembalik selain XOR [3] [8] [9].

Proses enkripsi algoritma TEA umumnya bekerja dalam *one cycle* (dua ronde) yang diulang sebanyak 32 kali (64 ronde), proses yang terjadi dalam *one cycle* dapat dilihat pada Gambar 2



Gambar 2 Proses enkripsi dalam satu putaran **[7]**

Proses yang terjadi dalam *one cycle* (dua ronde) adalah [9]:

1. Pergeseran *(shift)*
   v_0 dan v_1 masing-masing akan digeser sebanyak empat kali ke arah kiri dan lima kali ke arah kanan.
2. Penambahan
   Setelah v_0 dan v_1 mengalami pergeseran kemudian ditambahkan dengan kunci K_0-K_3, sehingga persamaan pada ronde 1 menjadi (1):

$$((v_0<<4) + K_0)\ ;\ ((v_0>>5) + K_1) \tag{1}$$

   dan persamaan pada ronde 2 adalah (2):

$$((v_1<<4) + K_2)\ ;\ ((v_1>>5) + K_3) \tag{2}$$

3. XOR ($\oplus$)
   Proses selanjutnya adalah melakukan operasi XOR pada v_0 , v_1 dan sum. Untuk masuk ke *cycle* selanjutnya, posisi v_0 dan v_1 akan ditukar, sehingga dalam satu *cycle* persamaannya menjadi (3):

$$\begin{aligned} v_0 &+= ((v_1<<4) + K_0) \oplus (v_1 + sum) \oplus ((v_1>>5) + K_1) \\ v_1 &+= ((v_0<<4) + K_2) \oplus (v_0 + sum) \oplus ((v_0>>5) + K_3) \end{aligned} \tag{3}$$

4. *Key schedule*
   Dalam algoritma ini, *Key* K_0 dan K_1 digunakan pada ronde ganjil sedangkan *Key* K_2 dan K_3 digunakan pada ronde genap.

Pseudo code proses enkripsi dapat dilihat pada Gambar 3

```
    TEA Encryption Process

1   uint32_t v0=v[0], v1=v[1], sum=0,i;
2   uint32_t delta=0x9E3779B9
3   uint32_t K0=K[0] K1=K[1] K2=K[2] K3=K[3]
4   for (i=0; i<32; i++)
5   {
6       sum+=delta;
7       v0+=((v1<<4)+ K0)XOR(v1 + sum)XOR((v1>>5) + K1)
8       v1+=((v0<<4)+ K2)XOR(v0 + sum)XOR((v0>>5) + K3)
9   }
10  v[0]=v0;v[1]=v1;
```

Gambar 3 Pseudo Code proses enkripsi

B. *Dekripsi TEA*

Dekripsi adalah proses mengembalikan dokumen ke bentuk semula agar dapat dibaca dan dimengerti sisi penerima. Dalam proses dekripsi diperlukan kunci yang sama yang digunakan pada proses enkripsi. Alur proses dekripsi pada algoritma TEA sama dengan yang terjadi pada proses enkripsi. Pada proses dekripsi, *key* yang digunakan merupakan kebalikan dari proses enkripsi. Semua round ganjil pada proses dekripsi menggunakan key K[1] kemudian K[0], sedangkan untuk round genap menggunakan key K[3] kemudian K[2] [6]. Sehingga persamaan proses dekripsi menjadi (4):

$$v_0 \mathrel{+}= ((v_0<<4) + K_2) \oplus (v_0 + sum) \oplus ((v_0>>5) + K_3)$$
$$v_1 \mathrel{+}= ((v_1<<4) + K_0) \oplus (v_1 + sum) \oplus ((v_1>>5) + K_1) \quad (4)$$

Pseudo code untuk proses dekripsi dapat dilihat pada Gambar 4

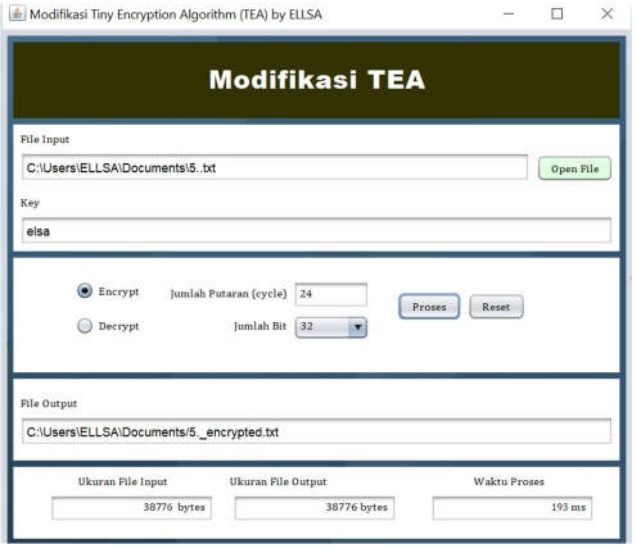
```
    TEA Decryption Process

1   uint32_t v0=v[0], v1=v[1], sum=0xC6EF3720, i;
2   uint32_t delta=0x9E3779B9
3   uint32_t K0=K[0] K1=K[1] K2=K[2] K3=K[3]
4   for (i=0; i<32; i++)
5   {
6       v1-=((v0<<4)+ K2)XOR(v0 + sum)XOR((v0>>5) + K3)
7       v0+=((v1<<4)+ K0)XOR(v1 + sum)XOR((v1>>5) + K1)
8       sum-=delta;
9   }
10  v[0]=v0;v[1]=v1;
```

Gambar 4. *Pseudo cod*e Proses Dekripsi

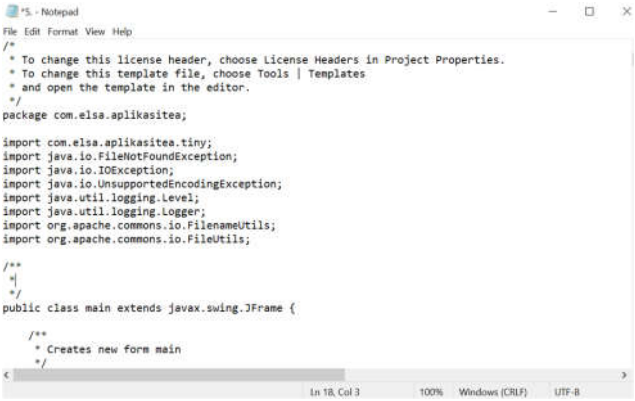
III. HASIL DAN PEMBAHASAN

Pada penelitian ini pengujian dilakukan dengan melakukan proses enkripsi dan dekripsi terhadap dokumen uji menggunakan sistem yang berhasil dirancang. Gambar 5 adalah tampilan sistem ketika dijalankan.



Gambar 5. Tampilan Sistem

Sistem yang dirancang memuat menu-menu sesuai dengan fungsionalitas yang diharapkan. Pengguna dapat memilih teknis enkripsi ataupun dekripsi beserta dengan modifikasi yang diinginkan. Ketika pengguna melakukan proses enkripsi maupun dekripsi, selain mendapatkan dokumen hasil pemrosesan, pengguna juga akan mendapatkan informasi mengenai proses yang telah dilakukan, seperti ukuran dokumen hasil pemrosesan dan waktu proses yang dibutuhkan. Gambar 6 merupakan salah satu dokumen yang diuji cobakan, sedangkan Gambar 7 merupakan hasil dari proses enkripsi.



Gambar 6. Contoh dokumen asli

Gambar 7. Dokumen hasil enkripsi

Dokumen yang telah melalui proses enkripsi berubah menjadi dokumen yang berisikan karakter-karakter yang tidak dapat dipahami oleh orang lain, untuk dapat membaca dokumen tersebut perlu dilakukan proses dekripsi menggunakan kunci yang sama dengan proses enkripsi.

Untuk dapat mengetahui pengaruh modifikasi algoritma TEA terhadap hasil enkripsi maupun dekripsi dilakukan pengamatan pada waktu pemrosesan dan perubahan ukuran dokumen hasil enkripsi menggunakan algoritma TEA yang telah dimodifikasi. Bentuk pengujian pada penelitian ini terbagi dalam tiga skenario: 1) Proses Enkripsi dan Dekripsi dengan Algoritma TEA Standar. 2) Modifikasi pada jumlah *cycle,* 3) Modifikasi pada jumlah *bit-input*. Proses enkripsi dan dekripsi akan menggunakan file dengan format PDF, docx dan txt.

A. Algoritma TEA standar

Pada skenario ini dilakukan proses Enkripsi dan Dekripsi pada 6 dokumen uji dengan ekstensi *pdf, docx,* dan *txt* menggunakan algoritma TEA standar. Pengujian dilakukan dengan mengamati perubahan ukuran dan waktu yang dibutuhkan sistem untuk melakukan proses enkripsi maupun dekripsi pada dokumen uji.

B. Modifikasi jumlah cycle

Pada skenario ini, modifikasi hanya dilakukan pada jumlah *cycle* yang dikerjakan pada proses enkripsi dan dekripsi sehingga *bit-input* dan panjang *key* masih tetap menggunakan aturan TEA. Masing-masing dokumen akan dienkripsi menggunakan algoritma TEA dengan iterasi sebanyak 202 *cycle* dengan panjang bit-input adalah 64-bit. Melalui skenario ini, didapatkan perubahan ukuran pada tiap-tiap dokumen setelah melalui proses enkripsi beserta dengan waktu yang dibutuhkan untuk proses enkripsi dan dekripsi dokumen tersebut.

C. Modifikasi jumlah bit-input

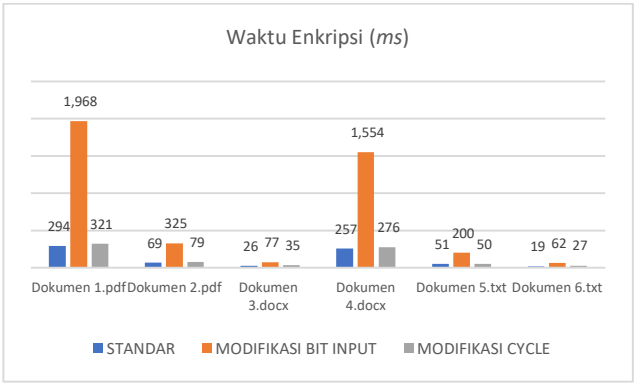
Skenario pengujian selanjutnya adalah mengubah jumlah bit-input dari yang algoritma TEA yang harusnya menerima 64-bit menjadi 32-bit *input*. Perubahan pada bit-input ini tidak mengubah konsep kerja dari algortima TEA. Algortima TEA akan tetap melakukan enkripsi dengan memecah bit-inputan menjadi dua blok dengan ukuran yang sama kemudian memprosesnya dalam 32 *cycle* (64 ronde).

Melalui tiga skenario pengujian yang telah dilakukan terhadap dokumen uji yang sama, masing-masing skenario memberikan hasil yang berbeda pada parameter yang diamati. Sistem membutuhkan waktu yang berbeda-beda untuk melakukan enkripsi pada tiap-tiap dokumen uji. Tabel I menunjukkan waktu yang dibutuhkan sistem untuk melakukan enkripsi dokumen uji menggunakan algoritma TEA standar, TEA Modifikasi jumlah bit-input, dan TEA Modifikasi jumlah perulangan.

TABEL I
PERBANDINGAN WAKTU ENKRIPSI ALGORITMA TEA STANDAR, MODIFIKASI BIT-INPUT, DAN MODIFIKASI PERULANGAN

Dokumen	Waktu Enkripsi (ms)		
	Standar	Modifikasi Bit Input	Modifikasi Cycle
Dokumen 1.pdf	294	1.968	321
Dokumen 2.pdf	69	325	79
Dokumen 3.docx	26	77	35
Dokumen 4.docx	257	1.554	276
Dokumen 5.txt	51	200	50
Dokumen 6.txt	19	62	27

Berdasarkan data di atas, algortima TEA standar membutuhkan waktu pemrosesan paling sedikit dibanding dengan dua bentuk modifikasi yang diajukan. Bentuk modifikasi pada jumlah perulangan terbukti membutuhkan waktu lebih sedikit dibandingkan dengan modifikasi pada jumlah bit-input. Gambar 8 merupakan representasi visual dari perbandingan waktu yang dibutuhkan untuk melakukan enkripsi menggunakan masing-masing skenario.



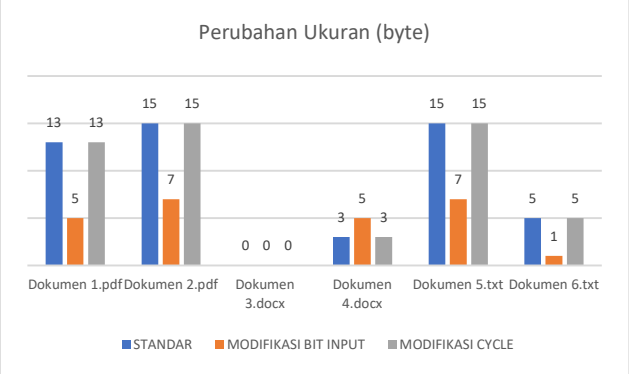
Gambar 8. Perbandingan Waktu Enkripsi

Untuk mengetahui perubahan ukuran yang terjadi pada dokumen uji setelah melalui proses enkripsi, pengamatan dilakukan pada ukuran dokumen hasil enkripsi hingga satuan *byte*. Tabel II merupakan data perubahan ukuran dokumen setelah melalui proses enkripsi menggunakan algortima TEA standar, modifikasi jumlah bit-input, modifikasi jumlah perulangan.

TABEL III
PERBANDINGAN PERUBAHAN UKURAN DOKUMEN SETELAH DIENKRIPSI MENGGUNAKAN ALGORITMA TEA STANDAR, MODIFIKASI BIT-INPUT, DAN MODIFIKASI PERULANGAN

Dokumen	Ukuran Asli	Ukuran Dokumen Enkripsi (*byte*)		
		Standar	Modifikasi Bit Input	Modifikasi Cycle
Dokumen 1.pdf	402.531	402.544	402.536	402.544
Dokumen 2.pdf	69.089	69.104	69.096	69.104
Dokumen 3.docx	14.864	14.864	14.864	14.864
Dokumen 4.docx	335.083	335.080	335.088	335.080
Dokumen 5.txt	38.769	38.784	38.776	38.784
Dokumen 6.txt	8.797	8.792	8.796	8.792

Melalui data hasil uji, algortima TEA yang dilakukan modifikasi terhadap jumlah bit-input ternyata menghasilkan perubahan ukuran yang paling kecil dibandingkan dengan algortima TEA standar maupun modifikasi jumlah perulangan. Sedangkan dokumen hasil enkripsi menggunakan algoritma TEA standar dan modifikasi jumlah perulangan menghasilkan dokumen dengan ukuran yang sama. Perbandingan perubahan ukuran dokumen dapat diamati dalam bentuk grafik pada Gambar 9.



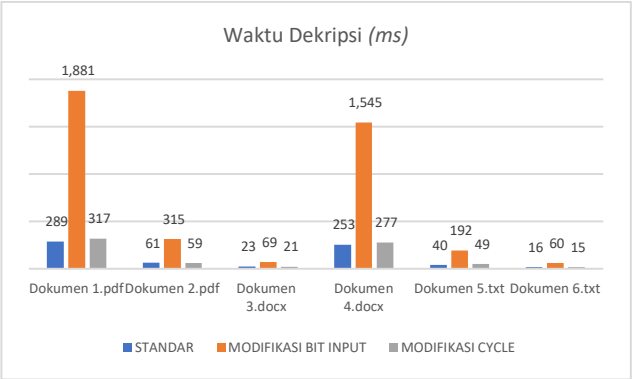
Gambar 9. Perbandingan Perubahan Ukuran Dokumen

Parameter yang diamati selanjutnya adalah waktu yang dibutuhkan sistem untuk melakukan dekripsi menggunakan algoritma TEA standar, TEA Modifikasi jumlah bit-input, dan TEA modifikasi jumlah perulangan. Tabel III berisikan data hasil pengamatan pada waktu dekripsi yang dibutuhkan dalam 3 skenario uji.

TABEL IIIII
PERBANDINGAN WAKTU ENKRIPSI ALGORITMA TEA STANDAR, MODIFIKASI BIT-INPUT, DAN MODIFIKASI PERULANGAN

Dokumen	Waktu Dekripsi (*ms*)		
	Standar	Modifikasi Bit Input	Modifikasi Cycle
Dokumen 1.pdf	289	1.881	317
Dokumen 2.pdf	61	315	59
Dokumen 3.docx	23	69	21
Dokumen 4.docx	253	1.545	277
Dokumen 5.txt	40	192	49
Dokumen 6.txt	16	60	15

Pada pengamatan waktu yang dibutuhkan untuk melakukan dekripsi, algortima TEA dengan modifikasi jumlah bit-input memakan waktu paling banyak jika dibandingkan dengan dua skenario uji lainnya. Sedangkan algoritma TEA standar dan Modifikasi jumlah cycle memiliki perbedaan waktu yang tidak terpaut jauh. Gambar 10 merupakan grafik perbandingan waktu yang dibutuhkan sistem untuk melakukan dekripsi pada dokumen uji menggunakan 3 skenario yang telah dibuat.



Gambar 10. Perbandingan Waktu Dekripsi

IV. KESIMPULAN

Berdasarkan hasil pengujian pada sistem yang dibangun menggunakan algoritma enkripsi modifikasi *Tiny Encrytion Algorithm* untuk mengamankan data berupa file *text* dapat ditarik kesimpulan sebagai berikut:

1. Sistem yang dirancang berhasil melakukan enkripsi dan dekripsi pada dokumen uji dengan ekstensi *pdf, docx,* dan *txt*.
2. Bentuk modifikasi pada algortima TEA berupa perubahan jumlah putaran dan ukuran bit-input berpengaruh terhadap waktu yang dibutuhkan sistem untuk melakukan pemrosesan dokumen. Selain itu, modifikasi terhadap algortima TEA juga berpengaruh terhadap ukuran dokumen hasil enkripsi.

UCAPAN TERIMA KASIH

Rasa syukur dan terima kasih kepada Tuhan Yang Maha tak terbatas, untuk izin dan segala bantuan-Nya sehingga penelitian dapat terlaksana dan terselesaikan dengan baik. Untuk orang tua dan segenap keluarga yang menjadi sumber dukungan dan doa yang tak terbatas siang maupun malam. Untuk jajaran dosen dan staff Jurusan Teknik Informatika Unesa atas segala bimbingan dan pendampingannya hingga selesai masa studi. Semoga penelitian ini menjadi ilmu yang bermanfaat untuk semua pembaca.

REFERENSI

[1] M. Khalil, "Real-Time Encryption/Decryption of Audio Signal," *International Journal of Computer Network and Information Security,* vol. 2, pp. 25-31, 2016.

[2] N. Aminudin, A. Maseleno, S. K, S. Hemalatha, K. S. kumar, Fauzi, R. Irviani dan M. Muslihudin, "Nur Algorithm on Data Encryption and Decryption," *International Journal of Engineering & Technology,* pp. 109-118, 2018.

[3] T. Asprina, M. Yamin dan Sutardi, "Pembangunan Aplikasi Keamanan Pesan Chatting dengan Menerapkan Algortima Tiny Encryption Algorithm berbasis Client Server," *SemanTIK,* vol. 4 (2), pp. 57-64, 2018.

[4] Liana, Sutardi dan N. F. Muchlis, "Aplikasi Enkripsi dan Dekripsi Data menggunakan Tiny Encryption Algorithm (TEA) berbasis Java," *semanTIK,* vol. 4 (1), pp. 39-48, 2018.

[5] A. Husna, J. M. Hulu dan Y. S. Hondro, "Implementasi Algoritma Tiny Encyrption Algorithm untuk Pengamanan File PDF," dalam *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)* , Medan, 2020.

[6] M. F. Mulya dan N. Rismawati, "Analisis dan Simulasi Algoritma TEA Untuk Enkripsi dan Dekripsi Pesan Text Menggunakan Cryptool2," *Jurnal Sistem Komputer dan Kecerdasan Buatan,* vol. 3 (1), pp. 31-38, 2019.

[7] Wikipedia, "Tiny Encryption Algorithm," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Tiny_Encryption_Algorithm#:~:text=In%20cryptography%2C%20the%20Tiny%20Encryption,a%20few%20lines%20of%20code.&text=The%20cipher%20is%20not%20subject%20to%20any%20patents.[Diakses 15 Januari 2021].

[8] Siswanto, M.Anif dan U. Abshor, "Pengamanan Data User Login dengan Algortima Kriptiografi TEA dan Notifikasi SMS," dalam *Prosiding Seminar Nasional SISFOTEK*, Padang, 2018.

[9] T. O. Purba dan A. S. Sembiring, "Perancangan Aplikasi Pengamanan Data Text dengan Menggunakan Algoritma Simetri TEA (Tiny Encryption Algorithm)," *KAKIFIKOM,* vol. 1 (2), pp. 59-66, 2019.