Implementasi Algoritma Kriptografi Tiny Encryption Algorithm Pada Keamanan Komunikasi Dan Data *Smart Card*

M R Adiwiganda[1], E Ariyanto[2], R Yasirandi[3]

Fakultas Informatika, Universitas Telkom, Bandung, Indonesia
[1]rizkiawenk@students.telkomuniversity.ac.id, [2]endroa@telkomuniversity.ac.id,
[3]batanganhitam@telkomuniversity.ac.id

Abstrak

***Smart Card* adalah alat yang digunakan untuk transaksi data dengan cepat dan praktis yang berbentuk chip yang ditanamkan pada kartu. Kartu tersebut sangat membantu proses identifikasi dan administrasi seperti kartu pasien pada rumah sakit terutama pada keadaan darurat. Data pasien hanya dapat diakses oleh orang tertentu oleh karena itu dibutuhkan mekanisme keamanan seperti algoritma kriptografi untuk mengamankan data pada kartu tersebut. Algoritma yang dibutuhkan untuk kartu pasien adalah algoritma yang memiliki kemampuan proses yang cepat tingkat keamanan yang tinggi. Pada paper ini telah diimplementasikan algoritma *Tiny Encryption Algorithm* pada *Smart Card* dan dihitung waktu proses serta tingkat keamanannya berdasarkan *Avalance Effect*.**

Berdasarkan dari hasil pengujian pada penelitian ini waktu proses enkripsi dan dekripsi data pasien dengan jumlah 600 baris riwayat pasien yang berukuran 30,6 Kb menghasilkan rata-rata waktu enkripsi sebesar 20,17 ms sedangkan dekripsi dibutuhkan waktu sebesar 17,21 ms yang menunjukkan waktu proses yang sangat kecil. Pengujian Avalance Effect yang dilakukan pada proses enkripsi TEA menghasilkan *Avalance Effect* sebesar 77% yang menunjukkan data yang dienkripsi memiliki hasil *chipertext* yang sangat acak sehingga isi data terjaga kerahasiaannya dengan tingkat keamanan yang tinggi. Hasil pengujian waktu proses dan AE tersebut dapat disimpulkan bahwa TEA merupakan algoritma yang memiliki proses mengamankan data yang cepat dengan keamanan yang tinggi dan waktu yang dapat ditoleransi oleh user *smart card*.

Kata kunci : *Smart Card*, *TINY ENCRYPTION ALGORITHM*, Enkripsi, Dekripsi.

Abstract

Smart Card is a tool that is used for fast and practical data transactions in the form of chips embedded in the card. This card is very helpful for the process of identification and administration such as patient cards in hospitals, especially in emergencies. However, patient data can only be accessed by certain people, therefore the card requires a security mechanism such as cryptographic algorithms to secure the contents of the data. Therefore the algorithm needed for the patient card is an algorithm that has fast processing capabilities and a high level of security. In this paper, we try to implement the Tiny Encryption Algorithm on a Smart Card and calculate the processing time and measure the level of security using the Avalance Effect.

Based on the results in this study the time of encryption and decryption of patient data with a total of 600 lines of patient medical history measuring 30.6 Kb resulted in an average encryption time of 20.17 ms while decryption required time of 17.21 ms that indicate processing time is very small. The Avalance Effect test performed on the TEA encryption process produces an Avalance Effect of 77% which shows that encrypted data has very random ciphertext results so that the data content is kept confidential with a high level of security. The test results of process time and AE can be concluded that the TEA is an algorithm that can process a fast data securing with high security and time that can be tolerated by smart card users.

Keywords: *Smart Card*, TINY ENCRYPTION ALGORITM, Encryption, Decryption.

1. Pendahuluan

Latar Belakang

Smart Card merupakan teknologi yang mempermudah transaksi data dan menurut sumber *Smart Card* adalah sebuah kartu plastik yang ditanam chip komputer dan dapat mengirim data tersebut ke perangkat – perangkat dan sesama pengguna[1]. Penggunaan *Smart Card* sudah banyak diimplementasikan pada kegiatan transaksi data maupun uang karena cara penggunaannya yang mudah, cepat dan tidak memakan banyak tempat. *Smart Card* sangat membantu banyak kegiatan manusia terutama kegiatan pendaftaran atau registrasi dan identifikasi diri pada suatu tempat seperti rumah sakit. *Smart Card* dalam bentuk kartu pasien dapat membantu proses administrasi pasien lebih cepat dan praktis terutama pada saat keadaan darurat. Isi data dari kartu pasien merupakan hal yang

harus dirahasiakan dan harus diamankan agar data atau informasi pasien yang tersimpan pada *Smart Card* tidak dapat dibaca dan diubah.

Oleh karena itu *Smart Card* harus memiliki algoritma keamanan data tinggi dan waktu proses yang cepat agar data dapat diterima dengan cepat pada keadaan darurat. Oleh karena itu algoritma kriptografi yang sesuai dengan kebutuhan ini adalah algoritma Tiny Encryption Algorithm (TEA) yang merupakan salah satu kandidat algoritma tercepat dengan tingkat keamanan yang tinggi seperti hasil penelitian [7] yang menguji performansi beberapa algoritma seperti AES, DES, DESL, HIGHT, SEA, dan TEA. Penelitian ini bertujuan untuk mengimplentasikan Tiny Encrytion Algorithm pada *Smart Card* sebagai algoritma enkripsi dekripsi data, menganalisis kecepatan waktu proses dan tingkat keamanan data.

Topik dan Batasannya

Pada penelitian ini akan diimplementasikan TEA pada *Smart Card* untuk mengamankan data agar data tidak mudah untuk dibaca dan diubah oleh pihak lain. Lalu algoritma ini akan dianalisis kemampuannya dalam mengamankan data tersebut. Batasan dari topik ini adalah ukuran data yang dapat disimpan pada *Smart Card* yang digunakan hanya berukuran 32Kb.

Tujuan

Penelitian ini bertujuan untuk mengimplementasikan TEA pada *Smart Card* sebagai algoritma enkripsi dan dekripsi data dan menganalisis kecepatan waktu proses dan tingkat keamanan data.

Organisasi Tulisan

Organisasi penulisan pada tugas akhir ini disusun dengan beberapa bagian sebagai berikut: Subbab 1 berisikan penjelasan singkat tentang topik dari tugas akhir ini. Subbab 2 berisikan penjelasan teori – teori yang digunakan pada penelitian ini. Subbab 3 berisikan rancangan sistem dan penjelasan alur sistem yang sudah dibangun. Subbab 4 berisikan hasil evaluasi dari sistem beserta analisisnya. Subbab 5 berisikan hasil kesimpulan yang didapat dari analisis sistem.

2. Studi Terkait

2.1 *Smart Card*

Banyak teknologi pada saat ini yang digunakan untuk mentrasfer data dan salah satu teknologi tersebut adalah *Smart Card*. Fungsi dari kartu ini adalah untuk menyimpan data yang digunakan untuk transaksi data yang dapat dengan mudah dibawa dan praktis. Walaupun implementasi *Smart Card* sudah banyak diterapkan akan tetapi *Smart Card* tidak sama dikarenakan setiap *developer* menggunakan standar pemograman dan struktur data yang berbeda[2].

Smart Card contact memiliki standar infrastruktur komunikasi tersendiri yang terdapat pada ISO/IEC 7816. Semua komunikasi antara *Smart Card* dan *reader*/komputer dilakukan dengan pertukaran *Application Protocol Data Unit* (APDU) dimana *reader* akan mengirim perintah dalam bentuk command APDU dan kartu akan merespon dengan response APDU. *Command* APDU memiliki format wajib berukuran 4-byte yaitu CLA, INS, P1, P2 dan sisanya berisikan data. Sedangkan response APDU memiliki format yang berbeda yaitu: DATA, SW1 dan SW2 dimana format ini menunjukkan hasil proses yang dilakukan oleh kartu sebagai contoh balasan menandakan fungsi berhasil dilakukan adalah 0x9000 (SW1:90 SW2:00). Berikut penjelasan singkat dari format *command* dan *response* APDU:

Tabel 1. Deskripsi Format *Command* APDU dan *Response* APDU

Format *Command* APDU	
Command	**Deskripsi *Command***
CLA	Instruksi class
INS	Instruksi fungsi spesifiki
P1	Parameter fungsi
P2	Parameter fungsi
Lc	Panjang data dikirim
Data	Data yang dikirim
Le	Panjang data yang diharapkan dari response

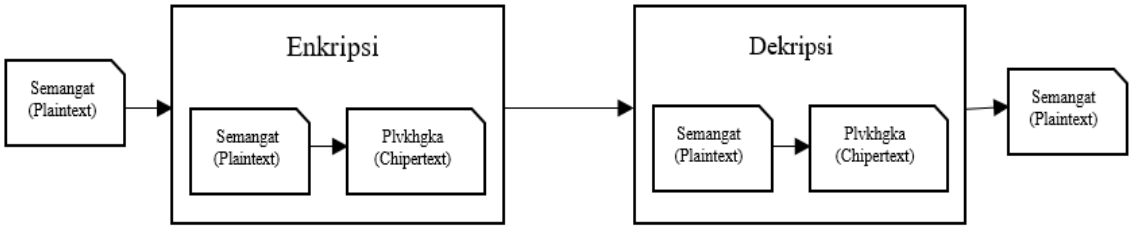
Format *Response* APDU	
Command	**Deskripsi *Command***
DATA	Balasan data hasil dari proses kartu
SW1	Status proses APDU
SW 2	Status proses APDU

2.2 *Java Card* dan *Eclipse Oxygen Framework*

Java Card merupakan salah satu bahasa pemrograman yang digunakan pada *Smart Card*. Program yang ditanamkan pada *Smart Card* ini menggunakan bahasa pemrograman Java yang dapat dijalankan oleh *Smart Card* dan sistem operasi yang digunakan oleh *Smart Card* adalah Java Card Virtual Machine (JCVM) [8]. *Framework* adalah sebuah software untuk membantu programmer untuk membangun program pada desktop. Untuk membangun program pada *Smart Card framework* yang digunakan adalah *Eclipse Oxygen Framework* yang mana *framework* ini menggunakan java sebagai bahasa pemrogramannya.

2.3 Kriptografi

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi[3]. Tujuan dari kriptografi adalah untuk menjaga kerahasiaan informasi pada saat transaksi sehingga informasi hanya dapat diketahui oleh pihak yang diperbolehkan. Konsep yang digunakan untuk mengubah dan membaca informasi tersebut adalah teknik enkripsi dan dekripsi. Pada pemrograman teknik kriptografi juga digunakan untuk merahasiakan informasi atau data agar aman dan dapat digunakan untuk transaksi data, konsep tersebut adalah algoritma kriptografi. Berikut ilustrasi dari teknik kriptografi:



Gambar 1. Ilustrasi Teknik Kriptografi

Terdapat dua jenis algoritma kriptografi berdasarkan kuncinya yaitu *Asymmetric Key Algorithm* yang mana algoritma ini menggunakan dua kunci yaitu *public key* dan *private key* yang digunakan pada proses enkripsi dan dekripsi, dan *Symmetric Key Algorithm* yang menggunakan satu kunci yang bersifat rahasia sehingga tidak semua dapat mengaksesnya.

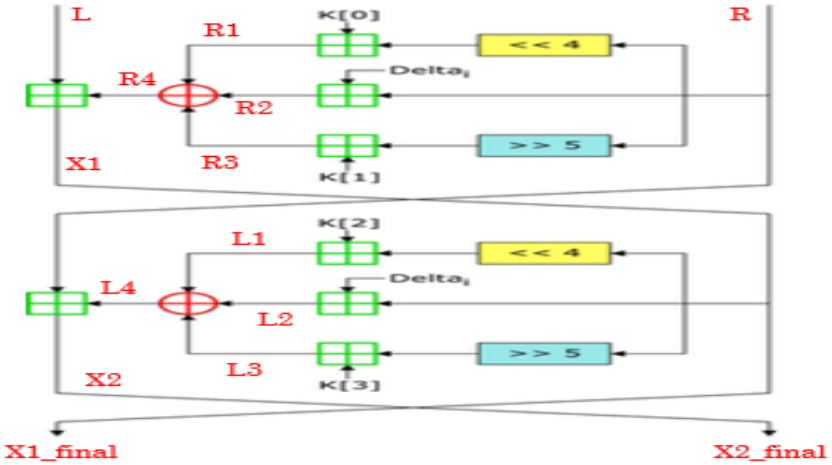
2.4 Avalance Effect (AE)

Avalance Effect (AE) adalah sebuah metode yang banyak digunakan pada algoritma enkripsi untuk melihat perubahan kecil pada *plaintext* atau kunci dapat menghasilkan perubahan signifikan pada *chipertext*[10]. AE banyak digunakan untuk mengukur kemampuan algoritma enkripsi dengan cara menghitung perubahan bit *plaintext* pada *chipertext*. Semakin banyak perubahan bit yang terjadi pada *plaintext* menunjukkan tingkat keamanan yang kuat dari algoritma enkripsi tersebut dan perubahan bit pada *chipertext* melebihi 50% menunjukkan tingkat keamanan yang tinggi [10] . Berikut rumus untuk menghitung AE[6]*:*

$$Avalance\ Effect = \frac{Jumlah\ bit\ berubah}{Jumlah\ bit\ total} * 100\% \quad (1)$$

2.5 Tiny Encryption Algorithm (TEA)

Tiny Encryption Algorithm (TEA) adalah algoritma kriptografi Simmetric key algoritma yang memiliki proses yang simple, penggunaan memori yang kecil dan tingkat keamanan yang baik dengan mengulang-ulang operasi matematika yang mudah. TEA diciptakan oleh David Wheeler and Roger Needham dari Cambridge University dan publikasikan pada tahun 1994[4]. Berikut skema jaringan fiestel yang digunakan pada algoritma TEA:



Gambar 2. Skema Jaringan Fiestel Pada TEA

TEA didesain untuk meminimalisir jejak memori (*memory footprint*) dan memaksimalkan kecepatan proses dengan membuat operasi dasar yang lemah dan sangat simple sementara keamanan yang tinggi dengan mengulang-ulang operasi tersebut[5]. Cara kerja TEA adalah sebagai berikut:

1. Proses Enkripsi TEA

Proses ini dilakukan dengan menggunakan *plaintext* sebesar 62-bit yang dibagi menjadi dua 32-bit input data yaitu L dan R. Lalu proses enkripsi ini menggunakan kunci sebesar 128-bit yang dibagi menjadi empat kunci dan kunci ini juga digunakan sebagai input data. Pertama data L akan digeser ke kiri sebanyak 4 bit lalu ditambahkan dengan kunci pertama yaitu K[0] dan data disimpan pada L2. Lalu data L di awal ditambahkan dengan data yang disebut *"Golden Ratio"* yang mana berfungsi untuk memastikan setiap hasi proses enkripsi dan dekripsi berbeda [9] dan hasilnya disimpan pada L3. Lalu data L di awal digeser ke kanan sebesar 5 bit lalu ditambahkan dengan kunci kedua yaitu K[1] dan disimpan pada L4. Lalu fungsi XOR digunakan pada data L2, L3 dan L4 yang artinya L2 di XOR dengan L3 dan hasilnya di XOR dengan L4 lalu hasil akhirnya disimpan pada L5. Setelah itu data L5 ditambahkan dengan data R yang disimpan pada X1. Pada saat ini proses enkripsi TEA sudah menjalankan setengah siklus.

Proses selanjutnya hampir sama dengan proses di atas akan tetapi data awal L diganti menjadi X1 dan kunci yang digunakan adalah K[2] dan K[3]. Hasil proses dengan menggunakan K[2] disimpan pada R2, K[3] disimpan pada R3 dan X1 yang ditambahkan dengan *"Golden Ratio"* disimpan pada R2. Lalu sama seperti proses sebelumnya R2, R3 dan R4 menjalankan fungsi XOR yang hasilnya disimpan pada R5 dan R5 ditambahkan dengan data L yang di awal dimana hasilnya disimpan pada X2. X1 dan X2 merupakan hasil akhir dari satu siklus enkripsi TEA yang mana proses enkripsi pada TEA dilakukan sebanyak 32 siklus untuk menghasilkan data X1 dan X2 terakhir dan disimpan pada X1_final dan X2_final. Terakhir kedua data digabung untuk mendapatkan hasil data yang terenkripsi.

2. Proses Dekripsi TEA

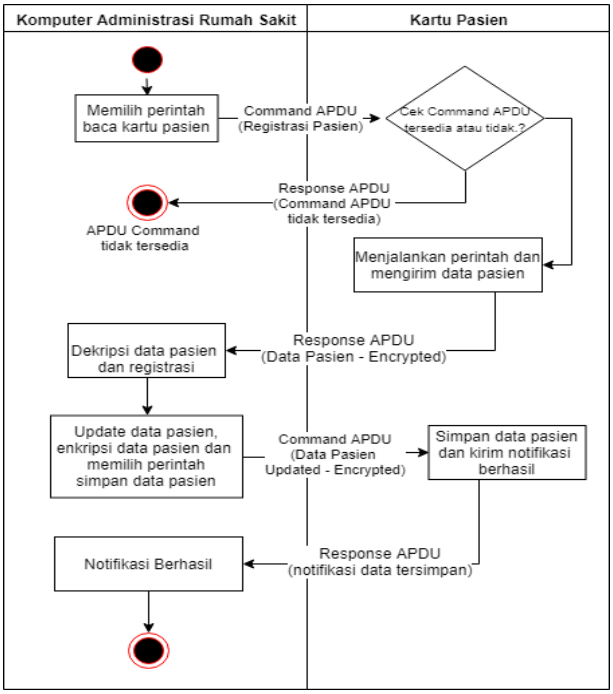
Proses ini tidak jauh berbeda dengan proses enkripsinya dimana alurnya prosesnya dibalik . Proses dekripsi dimulai dengan menjadikan data X1_final menjadi L dan X2_final manjadi R. Pertama data L akan digeser ke kiri sebanyak 4 bit lalu ditambahkan dengan kunci ketiga yaitu K[2] dan data disimpan pada L2. Lalu data L di awal ditambahkan dengan *"Golden Ratio"* yang mana hasilnya disimpan pada L3. Lalu data L di awal digeser ke kanan sebesar 5 bit lalu ditambahkan dengan kunci kedua yaitu K[3] dan disimpan pada L4. Lalu fungsi XOR digunakan pada data L2, L3 dan L4 yang artinya L2 di XOR dengan L3 dan hasilnya di XOR dengan L4 lalu hasil akhirnya disimpan pada L5. Setelah itu data L5 ditambahkan dengan data R yang disimpan pada X1. Pada saat ini proses enkripsi TEA sudah menjalankan setengah siklus.

Proses selanjutnya hampir sama dengan proses di atas akan tetapi data awal L diganti menjadi X1 dan kunci yang digunakan adalah K1 dan K2. Dimana hasil proses dengan menggunakan K[2] disimpan pada R2, K[3] disimpan pada R3 dan X1 yang ditambahkan dengan *"Golden Ratio"* disimpan pada R2. Lalu sama seperti proses sebelumnya R2, R3 dan R4 menjalankan fungsi XOR yang hasilnya disimpan pada R5 dan R5 ditambahkan dengan data L yang diawal dimana hasilnya disimpan pada X2. X1 dan X2 merupakan hasil akhir dari satu siklus enkripsi TEA yang mana proses enkripsi pada TEA dilakukan sebanyak 32 siklus untuk menghasilkan data X1 dan X2

terakhir dan disimpan pada X1_final dan X2_final. Terakhir kedua data digabung untuk mendapatkan hasil data yang terdekripsi.

3. Sistem yang Dibangun

Pada penelitian ini sistem yang dibangun adalah sistem administrasi rumah sakit dimana sistem ini akan menggunakan *Smart Card* sebagai kartu pasien yang berisikan data pasien. Data pasien tersebut akan disimpan pada kartu pasien dalam bentuk ciphertext yang dihasilkan dari implementasi algoritma kriptografi TEA. Berikut gambaran sistem yang dibangun:



Gambar 3. Rancangan Sistem

Pada sistem ini data yang akan diolah adalah data pasien yang mana data ini disimpan pada kartu dan data sudah terenkripsi dalam bentuk APDU. Untuk mendapatkan data yang ada pada kartu *reader*/komputer harus mengirim perintah dalam bentuk command APDU untuk dikonfirmasi bahwa data dapat kirim, setelah data dikirim selanjutnya data didekripsi untuk mendapatkan isi data yang asli agar dapat dibaca dan diproses. Proses yang dilakukan oleh *reader*/komputer hanya mengupdate data pasien dan selanjutnya data dienkripsi lagi untuk dikirim ke kartu. Pada setiap transaksi data yang dilakukan kartu dan *reader*/komputer data sudah dalam bentuk APDU. Data pasien yang digunakan pada sistem ini berisikan informasi umum yang dapat digunakan sebagai administrasi periksa di rumah sakit. Data tersebut juga menyimpan riwayat penyakit beserta penanganan dan obat yang diberikan. Pada penelitian ini data pasien dienkripsi dan dekripsi yang selanjutnya disimpan pada kartu dan dihitung lama waktu proses yang dibutuhkan beserta akan dihitung tingkat keamanannya.

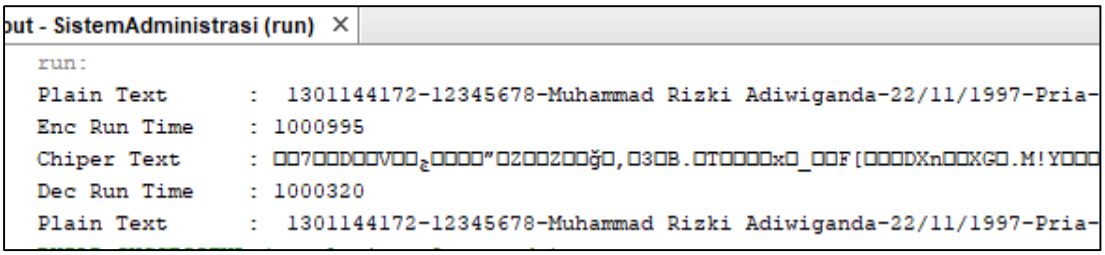
4. Evaluasi

Evaluasi bertujuan untuk mendapatkan kecepatan waktu enkripsi dan dekripsi dari Algoritma TEA dan tingkat keamanannya. Berikut penjelasan hasil pengujian, data pengujian dan analisis hasil pengujian. Pada penelitian ini hal yang dapat mempengaruhi hasil pengujian adalah spesifikasi dari mesin untuk memproses TEA yang digunakan, sedangkan kartu dan *reader* tidak mempengaruhi. Spesifikasi mesin yang digunakan pada penelitian ini adalah Processor Intel core i7-4510U CPU 2.00 GHz (4 CPU) ~ 2.6 GHz, RAM 4 GB dan Memory penyimpanan 500 GB. Berikut hasil pengujian yang dilakukan pada penelitian ini.

4.1 Hasil Pengujian Waktu Enkripsi dan Dekripsi

Pada pengujian algoritma enkripsi dan dekripsi TEA, dilakukan pengukuran kecepatan waktu prosesnya. Pengujian ini dilakukan dengan menggunakan empat data pasien yang dengan jumlah riwayat pasien dan ukuran yang berbeda (12 baris (800 byte), 60 baris (3,2 Kbyte), 194 baris (10,2 Kbyte) dan 600 baris (30,6

Kbytes)) untuk melihat pengaruhnya terhadap kecepatan waktu proses enkripsi dan dekripsi. Pada pengujian ini penghitungan waktu proses enkripsi dan dekripsi dilakukan dengan menjalankan sistem yang telah dibangun dan sistem akan menampilkan hasil penghitungan waktu proses yang dijalankan. Berikut contoh gambar hasil proses dan hasil pengujiannya:

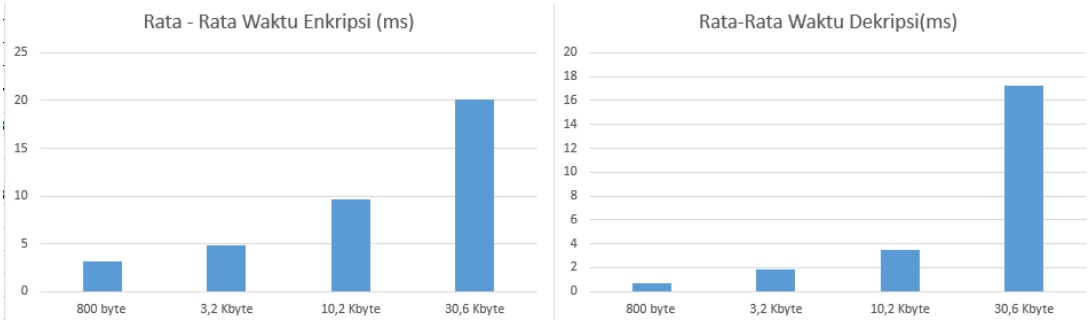


Gambar 6. Contoh Hasil Enkripsi dan Dekripsi Data Pasien

Tabel 2. Hasil Waktu Enkripsi dan Dekripsi (ms)

No	Ukuran	Proses	Percobaan Ke-										Rata-rata Waktu	
			1	2	3	4	5	6	7	8	9	10		
1	12 RP	enc	1,9	1,9	8,9	2,6	1,1	0,2	6,5	2,5	2,5	3,2	3,13	ms
		dec	0,6	1,1	1,4	0,8	0,6	0,3	0,2	0,1	1,1	0,6	0,68	ms
2	60 RP	enc	4,3	4,1	5,4	3,6	5,2	5,2	5,4	7,1	4,8	3,4	4,85	ms
		dec	1,2	1,6	1,7	1,7	0,3	1,4	5,3	1,7	1,7	1,7	1,83	ms
3	194 RP	enc	8,6	1,3	9,2	10,2	11,9	12,4	10,1	11,9	12,7	8,7	9,7	ms
		dec	3,7	9,3	0,3	3,8	1,3	3,7	3,4	4,8	0,4	4,5	3,52	ms
4	600 Rp	enc	23,3	19,4	17,2	20,1	18,9	21,5	19,7	20,7	18,3	22,6	20,17	ms
		dec	18,9	16,1	19,5	15,2	17,5	15,8	16,9	17,1	18,2	16,9	17,21	ms

RP : Jumlah riwayat pasien (Nama Penyakit, Penanganan, Tanggal, Obat) yang tersimpan



Gambar 7. Rata-Rata Waktu Enkripsi dan Dekripsi Berdasarkan Ukuran Data

Pengujian dilakukan 10 kali percobaan pada data yang sama untuk mendapatkan hasil rata-rata waktu yang dibutuhkan oleh proses enkripsi dan dekripsi TEA. Dengan menggunakan spesifikasi mesin diatas yang digunakan pada pengujian ini untuk melakukan proses enkripsi dan dekripsi TEA pada data pasien yang memiliki jumlah 600 baris riwayat pasien yang berukuran 30,6 Kbyte menghasilkan rata-rata waktu 20,17 ms sedangkan proses dekripsi nya sebesar 17,21 ms . Dari hasil pengujian ini menunjukkan bahwa TEA dapat melakukan proses enkripsi dan dekripsi dengan waktu yang sangat kecil sehingga TEA bagus untuk digunakan pada *smart card*.

4.2 Hasil Pengujian *Avalance Effect*

Pada pengujian ini proses enkripsi TEA akan diukur AE untuk melihat tingkat keamanannya. Pada pengujian hasil proses enkripsi TEA akan dibandingkan perubahan bit *chipertext* dengan bit chipertext yang bit pada kunci akan diubah-ubah per 1 bit. Semakin banyak perubahan bit yang terjadi maka akan semakin kuat tingkat keamanan yang dihasilkan TEA.Berikut contoh dan hasil pengujiannya :

```
Bit Key 1 : 010010101101
Bit Key 2:  011010101101
Chipertext 1: 1010 0011 0010 0010 0010 1010 1001 0010 1010 1010 0110 1010 0101
Chipertext 2: 1001 0100 0111 1010 1010 1110 1111 1010 0011 1000 0010 0101 1000
```

Gambar 8. Contoh Perubahan Bit.

Tabel 3. Hasil *Avalance Effect* TEA

Bit Chipertext		Jumlah Bit Key Diubah				
		1 bit	2 bit	3 bit	4 bit	5 bit
118	CTB	69	78	92	110	104
	AE	58%	66%	78%	93%	88%
103	CTB	64	102	92	68	75
	AE	62%	99%	89%	66%	73%
117	CTB	92	68	74	79	110
	AE	79%	58%	63%	68%	94%
97	CTB	70	94	64	87	71
	AE	72%	97%	66%	90%	73%
152	CTB	136	151	109	94	102
	AE	89%	99%	72%	62%	67%
Rata-rata Avalanche Effect		72%	84%	74%	76%	79%
Total rata-rata Avalanche Effect		77%				

CTB : Jumlah bit chipertext yang berubah
AE : Avalanche effect

Hasil pengujian ini menunjukkan jumlah perubahan bit yang terjadi pada *chipertext* saat bit pada kunci diubah-ubah sebanyak 1 bit. Chipertext yang digunakan pada pengujian memiliki panjang bit yang berbeda-beda sedangkan kunci yang digunakan memiliki panjang bit yang sama yaitu 128-bit dan kunci diubah sampai 5 bit untuk melihat AE pada TEA. Berdasarkan pengujian yang telah dilakukan TEA memiliki AE total rata-rata sebesar 77% dan algoritma yang memiliki AE lebih dari 50% menunjukkan hasil enkripsi dengan kemampuan keamanan yang tinggi.

5. Kesimpulan

TEA merupakan salah satu algoritma enkripsi dan dekripsi yang ringan, simple dan dengan tingkat keamanan yang baik. Berdasarkan dari hasil pengujian pada penelitian ini waktu proses enkripsi dan dekripsi data pasien dengan jumlah 600 baris riwayat pasien yang berukuran 30,6 Kb menghasilkan rata-rata waktu enkripsi sebesar 20,17 ms sedangkan dekripsi dibutuhkan waktu sebesar 17,21 ms yang menunjukkan waktu proses yang sangat kecil. Pengujian Avalance Effect yang dilakukan pada proses enkripsi TEA menghasilkan *Avalance Effect* sebesar 77% yang menunjukkan data yang dienkripsi memiliki hasil *chipertext* yang sangat acak sehingga isi data terjaga kerahasiaannya dengan tingkat keamanan yang tinggi. Hasil pengujian waktu proses dan AE tersebut dapat disimpulkan bahwa TEA merupakan algoritma yang memiliki proses mengamankan data yang cepat dengan keamanan yang tinggi dan waktu yang dapat ditoleransi oleh user *smart card*.

Daftar Pustaka

[1] Mohandes Mohamed. 2010. *A Smart Card Management and Application System.* Saudi Arabia. King Fahd University of Petroleum & Minerals.

[2] Setyoko Yoso, Nugraha I.G.B. 2014. *Multipurpose Smart Card System.* ICT For Smart Society (ICISS). 2014. Bandung. Institut Teknologi Bandung.

[3] Safrina Adilah 2017. *Implementasi Kriptosystem menggunakan metode Algoritma ECC dengan Fungsi Hash SHA-256 pada sistem ticketing online.* Bandung. Telkom University.

[4] Derek Williams. (2008). *The Tiny Encryption Algorithm(TINY ENCRYPTION ALGORITHM)*, Colombus State University.

[5] Ang Yee Hun, Stephanie & Md Naziri, Siti Zarina & Idris, Norina. (2012). The Development of Tiny Encryption Algorithm (TEA) Crypto-Core for Mobile Systems. International Conference on Electronic Devices, Systems, and Applications.

[6] Mandal Akash Kumar, Tiwari Archana. 2012. Analysis of Avalance Effect in Plaintext of DES Using Binary Codes. India. : Chhatrapati Shivaji Institute of Technology.

[7] Soren Rinne, Thomas Eisenbarth, Christof Paar. *Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers*. Germany : Horst Gortz Institute for IT Security.

[8] A Svenda Petr, Kur Jirl, Smolka Tobias. 2011. *Cryptographic Smart Cards & Java Card & PKI Tutorial*. Czech Republic: Masaryk University.

[9] Shepherd Simon. 2007. The Tiny Encryption Algorithm. USA : Taylor & Francis Inc.

[10] William Stallings. 2016. *Cryptography and network security : principles and practice* (Seventh ed.). Boston.