APLIKASI ENKRIPSI DAN DEKRIPSI DATA MENGGUNAKAN *TINY ENCRYPTION ALGORITHM* (TEA) BERBASIS JAVA

Liana*[1], Sutardi[2], Nur Fajriah Muchlis[3]
[*1,2,3]Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail: ***[1]lianait8@gmail.com.**, [2]sutardi_hapal@yahoo.com, [3]nurfajriah.muchlis@gmail.com.

Abstrak

Kebutuhan akan keamanan data semakin meningkat seiring dengan berkembangnya teknologi digital. Hampir semua kegiatan komunikasi dan transaksi sekarang dilakukan secara digital sehingga kekhawatiran akan adanya data yang hilang semakin menjadi. Data merupakan hal yang sangat penting, sehingga keakuratan dan ketepatan data sangatdibutuhkan dalam proses pengambilan keputusan. Pentingnya nilai dari suatu data menyebabkan harus tersedianya keamanan data, agar data jatuh secara tepat dan akurat pada pihak yang tepat.

Pada penelitian ini dirancang sebuah aplikasi keamanan data. Metode enkripsi digunakan untuk mengamankan data dengan menerapkan algoritma Tiny Encryption Algorithm (TEA). Pada proses enkripsi dilakukan 16 ruond dimana setiap round terdiri dari; penambahan, subtitusi kunci, dan data, serta XOR.

Hasil dari penelitian dibangun suatu aplikasi kkeamanan data yang dapat mengamankan data dengan menerapkan algoritma Tiny Encryption Algorithm (TEA). Dari hasil pengujian membuktikan aplikasi mampu mengamankan data dengan tipe data yang berekstensi txt dan docx. Semakin besar ukuran dari data yang dienkripsi maka semakin lamawaktu yang dibutuhkan untuk proses enkripsi.

Kata kunci—Data, TEA, Enkripsi, Dekripsi.

Abstract

The need for data security is increasing along with the development of digital technology. Almost all communication and transaction activities are now done digitally so that worries about missing data will increase. Data is very important, so accuracy and precision of data is needed in the decision-making process. The importance of the value of a data causes the availability of data security, therefore data falls precisely and accurately to the right party

In this study designed a data security application. Encryption method is used to secure data by applying algorithm of Tiny Encryption Algorithm (TEA). In the encryption process performed 16 rounds where each round consists of; additions, key substitutions, and data, and XOR.

The results of the study built a data security application that can secure data by applying the algorithm Tiny Encryption Algorithm (TEA). From the test results prove that application is able to secure extension txt and docx. The larger the size of the encrypted data the more time it takes for the encryption process.

Keywords*— Data, TEA, Encryption, Decryption*

1. PENDAHULUAN

Kebutuhan akan keamanan data semakin meningkat seiring dengan berkembangnya teknologi digital. Hampir semua kegiatan komunikasi dan transaksi sekarang dilakukan secara digital sehingga kekhawatiran akan adanya data yang hilang semakin menjadi. Data merupakan hal yang sangat penting, sehingga keakuratan dan ketepatan data sangat dibutuhkan dalam proses pengambilan keputusan. Penggunaan data sudah bukan untuk kepentingan suatu individu saja, sektor-sektor lain yang penting seperti perusahaan, lembaga masyarakat dan pemerintahan. Pentingnya nilai dari suatu data

menyebabkan harus tersedianya keamanan data, agar data jatuh secara tepat dan akurat pada pihak yang tepat.

Salah satu yang penting dalam komunikasi mengunakan komputer untuk menjamin kerahasiaan data adalah enkripsi. Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti. Enkripsi dapat diartikan sebagai kode atau *chiper*. sebuah sistem pengkodean menggunakan suatu *table* atau kamus yang telah didefenisikan untuk mengganti kata dari informasi. Sebuah kode menggunakan suatu algoritma yang dapat mengkodekan semua aliran data, bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti. Karena teknik *chipher* merupakan suatu sistem yang telah siap untuk diautomasi, maka teknik ini digunakan dalam sistem keamanan komputer dan jaringan. Enkripsi dibentuk berdasarkan suatu algoritma yang akan mengacak suatu informasi menjadi bentuk yang tidak bisa dibaca atau tidak bisa dilihat. Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikan informasi teraacak kebentuk semula.

Berbagai algoritma kriptografi yang sekarang ini telah dan sedang dikembangkan, salah satunya algoritma kunci simetris ataupun asimetris (pembagian berdasarkan kunci). Salah satu metode enkripsi data adalah *Tiny Encryption Algorithm* (TEA). *Tiny Encryption Algorithm* (TEA) merupakan suatu algoritma sandi yang diciptakan oleh David Wheeler dan Roger Needham dari *Computer Laboratory*, Cambridge University, Inggris pada bulan November 1994. Algoritma ini merupakan algoritma penyandian *block cihper* yang yang menggunakan proses *feistel network* dengan panjang kunci 128 bit, dengan cara memproses 64-bit input sekali waktu dan menghasilkan 64-bit output.

Dalam Penelitian ini akan membahas "Aplikasi Enkripsi dan Dekripsi Data Menggunakan *Tiny Encryption Algorithm* (Tea) Berbasis Java".

2. METODE PENELITIAN

2.1 Aplikasi Berbasis Desktop

Aplikasi *desktop* adalah aplikasi yang dapat berjalan secara sendiri atau independen dalam sistem *desktop* komputer atau laptop dan dapat menjalankan serangkaian aktivitas dengan diatur oleh pengguna. Pemilihan aplikasi berbasis *desktop* biasanya ditujukan kepada mereka yang memiliki koneksi internet yang kurang baik dan sangat peduli dengan keamanan sistem.

2.2 Data

Data adalah deskripsi tentang benda, kejadian, aktifitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh langsung kepada pemakai. Data dapat berupa nilai terformat, teks, citra, *audio* dan *video* [1].

2.3 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan data ketika data dikirim dari suatu tempat ke tempat yang lain. Dalam perkembangannya, *kriptografi* juga digunakan untuk mengidentifikasi pengiriman data dan tanda tangan digital dan keaslian data dengan sidik jari digital [2].

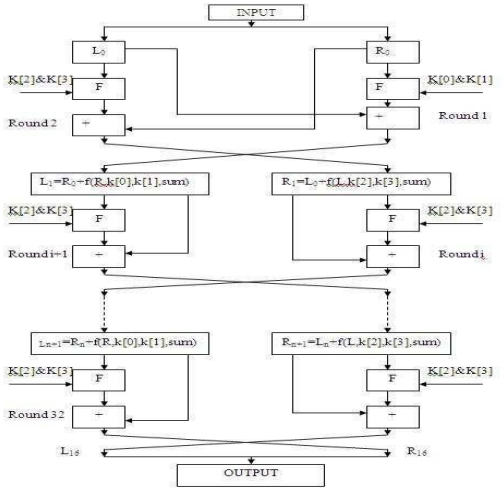
Kriptografi adalah suatu metode keamanan untuk melindungi suatu informasi dengan menggunakan kata-kata sandi yang hanya bisa dimengerti oleh orang yang berhak mengakses informasi tersebut. Kriptografi merupakan satu-satunya metode yang digunakan untuk melindungi informasi yang melalui jaringan komunikasi yang menggunakan *landline* (kabel di bawah tanah), satelit komunikasi, dan fasilitas *microwave* (gelombang mikro). Prosedur-prosedur kriptografi juga bisa digunakan untuk authentifikasi pesan, *digital signature*, dan identifikasi pribadi untuk mengotorisasi transfer uang secara digital melalui ATM, kartu kredit, dan melalui suatu jaringan [2].

2.4 Algoritma *Tiny Encryption 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. 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. 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 [3]. *Tiny Encryption Algorithm* (TEA) ditunjukkan pada Gambar 1.



Gambar 1. Algoritma TEA

Proses diawali dengan *input-bit* teks sebanyak 64-bit, kemudian 64-bit teks tersebut dibagi menjadi dua bagian, yaitu sisi kiri (L_0) sebanyak 32-bit dan sisi kanan (R_0) sebanyak 32-bit. Setiap bagian teks akan dioperasikan sendiri-sendiri. R_0 (*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 L_0 *(Y)* yang akan menjadi R_1.

Sisi sebelah kiri akan mengalami proses yang sama dengan sisi sebelah kanan. L_0 *(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 R_0 *(Z)* yang akan menjadi L_1.

Langkah-langkah penyandian dengan algoritma TEA dalam satu cycle (dua round) :

1. Pergeseran *(shift)*

   Blok teks 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 teks 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 Y_1 menjadi Z_1 dan Z_1 menjadi Y_1 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 cip*her lainnya. Yaitu pada prinsipnya adalah sama pada saat proses enkripsi. Hal yang berbeda adalah penggunaan teks 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).

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

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

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 : kuunci

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

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

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

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 : kuunci

2.5 Rational Unified Process (RUP)

Pengertian *Rational Unified Process* (RUP) menurut IBM adalah kerangka proses yang menyediakan simulasi sistem pada industri untuk sistem, software, implementasi, dan manajemen proyek yang efektif. RUP adalah salah satu dari sekian banyak proses yang terdapat di dalam *Rational Process Library*, yang memberikan simulasi terbaik untuk pengembangan atau kebutuhan proyek [4].

2.6 *Flowchart*

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong analis dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian [5].

2.7 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak [5].

Dalam terapannya, UML digambarkan dalam bentuk diagram. Diagram yang digunakan dalam penelitian, yaitu :

1. *Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi itu. Syarat penamaan pada use case adalah nama didefinisikan dengan sederhana dan mudah dipahami.
2. Diagram aktivitas menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.
3. Diagram sekuence menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Jumlah diagram sekuen yang harus digambar adalah minimal sama dengan jumlah *use case* yang didefinisikan.
4. Diagram class Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

2.8 Pemograman *Java*

Java sebagai bahasa pemrograman, java dapat membuat seluruh bentuk aplikasi, desktop, web dan lain-lain, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. *Java* adalah bahasa pemrograman berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source* [10].

3. HASIL DAN PEMBAHASAN

3.1 Gambaran Umum Aplikasi

Proses yang akan berjalan pada sistem ini akan dijelaskan sbagai berikut, pertama-tama pengguna akan memilih instruksi apa yang akan digunakan enkripsi atau dekripsi. Pada pilihan enkripsi pengguna memasukkan data (*plaintext*) yang kemudian akan diubah oleh sistem menjadi inti sari dan menampilkan identitas data. Selanjutnya pengguna memasukkan kunci, sistem akan melakukan *key expansion* dan menjalankan proses enkripsi yang akan menghasilakan data baru yang telah terenkripsi (*ciphertext*). Isi informasi pada data hasil enkripsi (*ciphertext*) tidak akan diketahui sebelum data (*ciphertext*) tersebut didekripsi. Dekripsi proses yang berlansung hampir sama dengan proses enkripsi. Data yang akan didekripsi adalah data yang telah dienkripsi oleh sistem berupa data *ciphertext* dan kunci (*key*) yang digunakan adalah (*key*) yang sama dengan masukan kunci (*key*) pada proses enkripsi.

3.2 Diagram *Flowchart*

Adapun *flowchart* gambaran umum aplikasi proses enkripsi ditunjukkan pada Gambar 2.

3.3 Flowchart Aplikasi Dekripsi

Adapun *flowchart* gambaran umum aplikasi proses dekripsi ditunjukkan pada Gambar 3.

3.4 Diagram *Flowchart Tiny Encryption Algorithm* (TEA)

1. Perancangan *Flowchart Enkripsi*

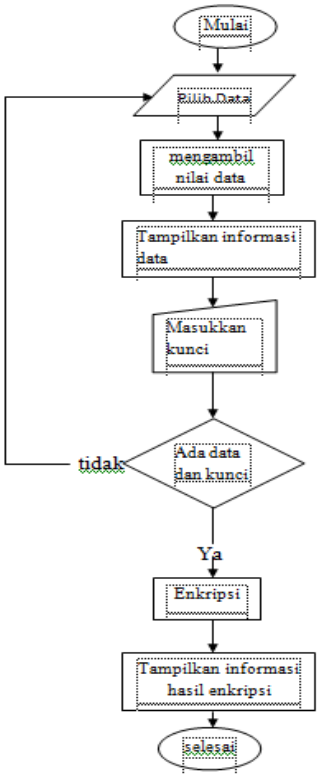
Flowchart proses enkripsi ditunjukkan pada Gambar 4.

2. Perancangan *Flowchart* Dekripsi

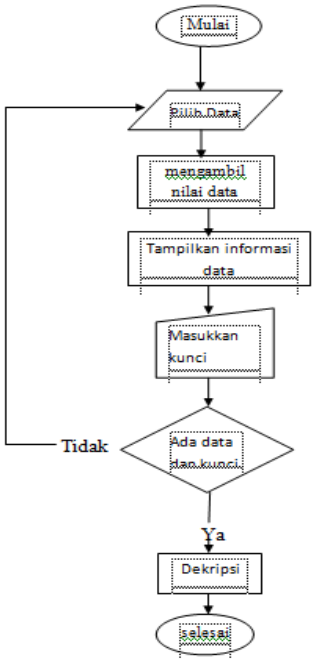
Flowchart proses dekripsi ditunjukkan pada Gambar 5.

3.5 Rancangan Sistem

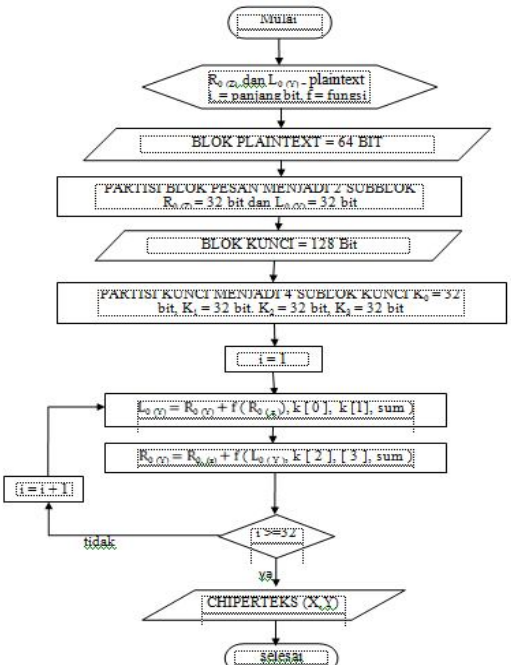
Unified Modelling Language (UML). Dalam laporan ini, penulis menyajikan rancangan sistem menggunakan 4 diagram, yaitu diagram *use case*, diagram *activity*, diagram *sequence*, dan diagram *class*.



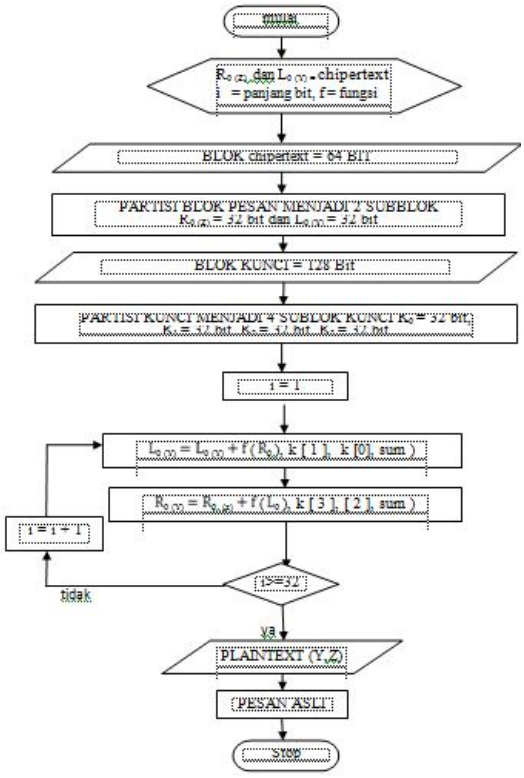
Gambar 2. Gambaran Umum Proses Enkripsi



Gambar 3. Gambaran Umum Proses Dekripsi

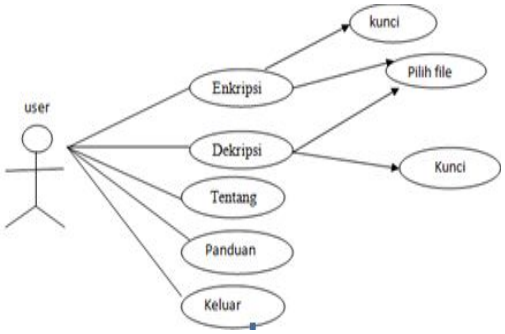


Gambar 4. *Flowchart* Enkripsi



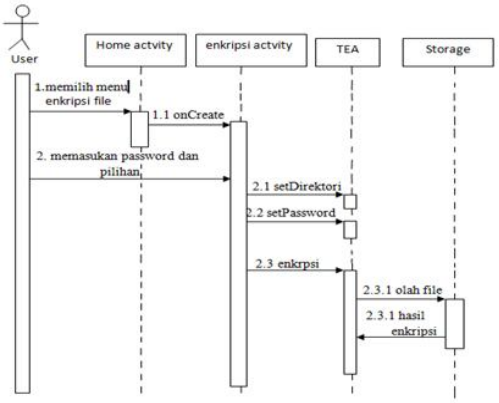
Gambar 5. *Flowchart* Dekripsi

1. Diagram *Use case* ditunjukkan oleh Gambar 6.



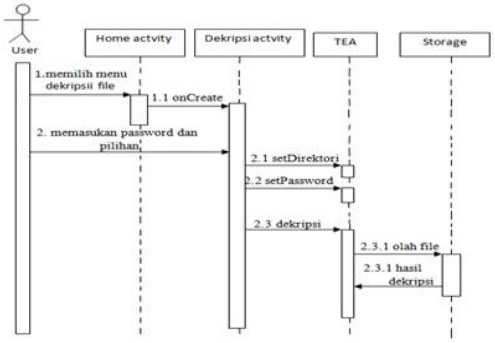
Gambar 6. Diagram *use case*

2. *Sequence Diagram* Enkripsi *File* ditunjukkan pada Gambar 7



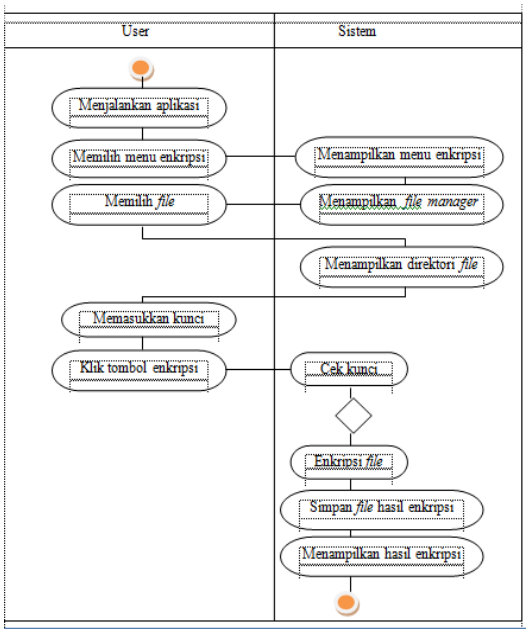
Gambar 7. *Sequence Diagram* Enkripsi *File*

3. *Sequence Diagram* Dekripsi *File* ditunjukkan pada Gambar 8.



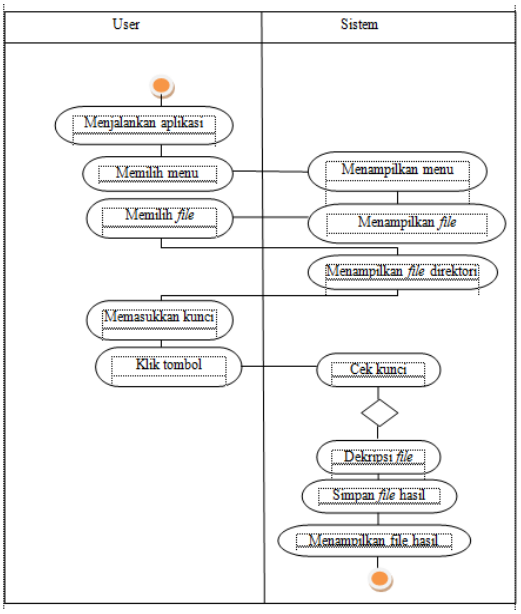
Gambar 8. *Sequence Diagram* Dekripsi *File*

4. *Activity Diagram* Enkripsi *File* ditunjukkan pada Gambar 9.



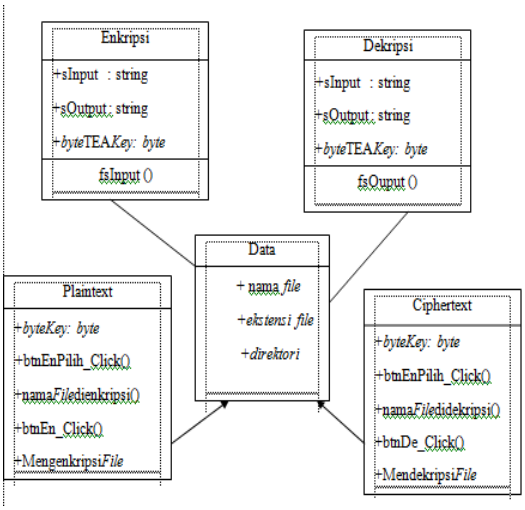
Gambar 9 *Activity Diagram* Enkripsi *File*

5. *Activity Diagram* Dekripsi *File* ditunjukkan pada Gambar 10.



Gambar 10. *Activity Diagram* Dekripsi *File*

6. *Class Diagram* ditunjukkan pada Gambar 11.



Gambar 11. *Class Diagram File*

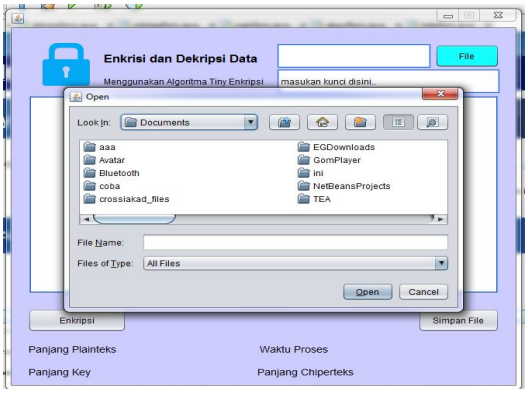
3.6 Implementasi

Aplikasi enkripsi dan dekripsi yang telah dirancang merupakan sebuah aplikasi yang menggunakan bahasa pemograman *Java NetBeans* yang dibangun untuk mengamankan data. Aplikasi ini menggunakan *Tiny Encryption Algorithm* (TEA).

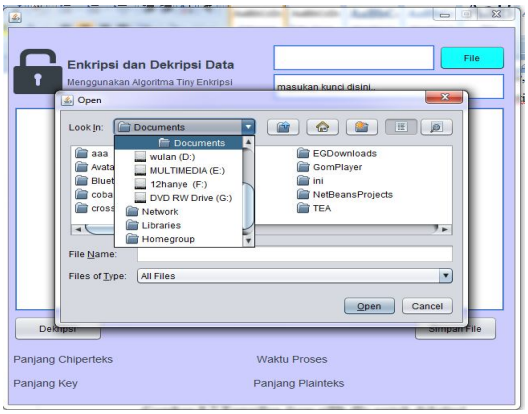
Menu utama merupakan *form* tampilan awal dari aplikasi enkripsi data dengan menggunakan *Tiny Encryption Algorithm* (TEA). Pada aplikasi ini terdapat beberapa menu lain yakni: enkripsi, dekripsi, panduan. Gambar 12 menunjukkan tampilan *Menu* Beranda.

Gambar 12 Tampilan Menu Beranda

Proses enkripsi ditunjukkan pada Gambar 13 dan Proses dekripsi ditunjukkan pada Gambar 14.



Gambar 13 Proses Enkripsi



Gambar 14 Proses Dekripsi

3.7 Pengujian

1. Pengujian Terhadap Waktu

Pada penelitian ini dibahas mengenai hasil dari enkripsi *file* dengan melakukan pengujian terhadap sampel dan meninjau bagaimana pengaruh ukuran terhadap waktu. Sampel yang digunakan memiliki tipe *file* yang sama, dengan kunci yang sama dan ukuran yang berbeda. Sampel yang digunakan pada pengujian ini sebanyak 10 sampel yakni; sampel 1.txt, sampel 2 txt, sampel 3 txt, sampel 4 txt dan sampel 5 txt. Tujuan dari pengujian ini untuk mengetahui hubungan antara ukuran *file* terhadap waktu,. Adapun hasil yang di dapat dari pengujian terdapat pada Tabel 1.

Tabel 1. Pengujian Terhadap Waktu

No	Nama	Ukuran	Kunci	Waktu
1	Sampel 1	95,0 byte	TEA	15 ms
2	Sampel 2	453,0 byte	TEA	31 ms
3	Sampel 3	1682,0 byte	TEA	88 ms
4	Sampel 4	2470,0 byte	TEA	130 ms
5	Sampel 5	3642,0 byte	TEA	150ms

2. Pengujian Terhadap Ukuran

Pada pengujian ini dilakukan proses enkripsi pada sampel 1.txt, sampel 2.txt, sampel 3.txt, sampel 4.txt, sampel 5.txt. Tujuan dilakukan pengujian ini untuk mengetahui perubahan ukuran dari *plaintext* menjadi *ciphertext*. Adapun hasil dari pengujian disajikan pada Tabel 2.

Tabel 2. Pengujian Terhadap Ukuran

No	Nama	Kunci	Ukuran (Byte)	
			Plainteks	*Ciphertext*
1	Sampel 1	TEA	95,0	96,0
2	Sampel 2	TEA	453,0	443,0
3	Sampel 3	TEA	1682,0	1604,0
4	Sampel 4	TEA	2470,0	2356,0
5	Sampel 5	TEA	3642,0	3542,0

4 KESIMPULAN

Berdasarkan penelitian dan hasil pengujian yang dilakukan terhadap aplikasi enkripsi dan dekripsi data menggunakan algoritma *Tiny Encryption*, maka kesimpulan yang didapat adalah sebagai berikut :

1. Aplikasi enkripsi dan dekripsi data berhasil dibangun untuk mengamankan berbagai jenis data bertipe *txt* dan *doc.*
2. Algoritma *Tiny Encryption* berhasil diimplementasikan dengan melalui beberapa tahap pengujian dalam sistem enkripsi dan dekripsi data dengan menggunakan kunci sepanjang 16 karakter atau lebih. Jika kurang dari 16 karakter algoritma *Tiny Encryption* tidak dapat mengamankan *file.*
3. Ukuran *file* sangat mempengaruhi lamanya proses enkripsi dekripsi *file*. Semakin besar ukuran *file* semakin lama proses yang diperlukan sistem untuk melakukan enkripsi *file*. .

5 SARAN

Adapun saran yang dapat diberikan untuk pengembangan dan perbaikan aplikasi ini untuk selanjutnya adalah diharapkan dapat

dilakukan pengembangan pada aplikasi enkripsi dan dekripsi ini, dengan menambahkan jenis *file* yang dienkripsi dan didekripsi seperti file bertipe pdf, jpeg, xlxs, video, dan audio.

DAFTAR PUSTAKA

[1] Kadir. 2011. Pengertiana data. Informatika Bandung. Bandung

[2] Ariyus. 2008. Defenisi Kriptografi. Informarika Bandung. Bandung.

[3] Mukti. 2014. "Kriptografi File Citra Menggunakkan Algoritma TEA". ITB. Bandung.

[4] Munawar, 2015, Pengembangan Aplikasi RUP, ,Graha Ilmu, Yogyakarta.

[5] Rosa, A.S., Salahuddin, M. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Penerbit Modula. Bandung