Perbandingan Implementasi Algoritma Blowfish dan Algoritma Tiny Encryption Algorithm (TEA) Dalam Aplikasi Enkripsi Pengaman File

Eka Arriyanti[1)], **Awang Harsa Kridalaksana**[2)], **Bagus Prasojo**[3)]

[1,2,3)] Jurusan Teknik Informatika, STMIK Widya Cipta Dharma Samarinda
Jl. M.Yamin No. 25, Samarinda, 75123
E-mail : ekaaryanti@wicida.ac.id[1)], bprasojjo1@gmail.com[3)]

ABSTRAK

Perbandingan implementasi algoritma *Blowfish* dan algrotima *Tiny Encryption Algorithm* (TEA) dalam aplikasi enkripsi pengaman file adalah suatu penelitian untuk menerapkan algoritma *Blowfish* dan TEA dalam pengaman data (file), yang dibandingkan berdasarkan waktu proses, kecepatan proses dan keamanan. Pengukuran waktu didapatkan dari lamanya waktu yang dibutuhkan oleh aplikasi untuk melakukan proses enkripsi dan dekripsi. Pengukuran kecepatan didapatkan dari pembagian antara ukuran dari file yang akan diproses dan waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi, Pengukuran keamanan didapatkan dari pengujian menggunakan *recovery software*.

Metode pengembangan sistem yang digunakan adalah *Rapid Application Development* (RAD). Alat bantu analisis dan desain adalah *Unified Modeling Language* (UML). Pengujian sistem yang digunakan adalah pengujian *white-box* dan *black-box*. Dari pengujian *white-box* dan *black-box* terhadap sistem, ditunjukkan bahwa aplikasi dapat berjalan untuk tujuan enkrisi dan dekripsi.

Hasil dari perbandingan implementasi kedua algoritma menunjukkan bahwa TEA lebih unggul dibandingkan *Blowfish* sebesar 1 : 1,5809 untuk kecepatan enkripsi, dan 1 : 1,4839 untuk kecepatan dekripsi. Dengan demikian, berdasarkan kecepatan, di mana kecepatan bergantung pada waktu proses, maka TEA lebih unggul dibanding *Blowfish*. Sedangkan dari segi keamananan, keduanya aman menurut *recovery software*.

Kata Kunci : *algoritma_Blowfish, algoritma_Tiny_Ecryption_Algorithm (TEA), pegaman_file*

---

1. PENDAHULUAN

Enkripsi merupakan salah satu alternatif dalam menjaga keamanan *file*. Hal ini dikarenakan enkripsi memungkinkan seseorang untuk mengubah data atau *file* yang memiliki informasi tertentu menjadi serangkaian data yang sulit atau mustahil untuk dimengerti dan diambil informasinya tanpa melakukan dekripsi. Masalah utama yang menjadikan enkripsi sangat penting untuk keamanan *file* adalah rentannya pencurian data yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Hal tersebut terjadi karena nilai dari informasi yang bisa didapatkan dari data itu sendiri dimana nilai dari informasi bisa sangat berharga dan penting bagi sebagian orang, sehingga mereka tidak ingin informasi tersebut diketahui oleh orang yang tidak berhak untuk mengetahuinya.

Untuk mengatasi masalah keamanan dan kerahasiaan data atau informasi yang sering terjadi, maka dilakukan penelitian tentang keamanan *file* untuk menjaga keamanan dan kerahasiaan *file* menggunakan dua algoritma *Block-chiper*, yaitu algoritma *Blowfish* dan *Tiny Encryption Algorithm* (TEA). Alasan pemilihan kedua algoritma tersebut adalah karena secara teoritis, keduanya cukup kuat untuk menjaga kerahasiaan dan keamanan *file*.

Penelitian ini akan membandingkan hasil enkripsi dan dekripsi dari penerapan algoritma *Blowfish* dan *Tiny Encryption Algorithm* dalam satu sistem yang dibangun, sehingga diketahui keunggulan antara keduanya.

2. RUANG LINGKUP PENELITIAN

Dalam Penelitian ini permasalahan menckup :

1. Cakupan Masalah
   Cakupan masalah penelitian ini dirumuskan sebagai berikut : “Bagaimanakah perbandingan aplikasi enkripsi menggunakan algoritma *Blowfish* dan *Tiny Encryption Algorithm* ?”
2. Batasan-batasan Penelitian
   Permasalahan dibatasi sebagai beritut :
   1) Proses enkripsi dan dekripsi dilakukan pada *file* secara satu persatu (tidak kolektif).
   2) Menggunakan algoritma Blowfish dan TEA dengan keluaran sistem yang berbeda.
   3) Perbandingan hasil enkripsi kedua algoritma dilakukan dalam hal perubahan pada *file*, waktu proses, dan kecepatan proses.
   4) Uji keamanan menggunakan *Software Recovery*.
3. Rencana Hasil yang didapat
   Berdasarkan rumusan dan batasan masalah, maka dapat disimpulkan rencana hasil yang didapat adalah melakukan perbandingan antara algoritma *Blowfish* dan *Tiny Encryption Algorithm* (TEA), sehingga diketahui kelebihan dari masing-masing algoritma.

3. BAHAN DAN METODE

3.1 Penjelasan Bahan

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu. Jadi aplikasi itu bisa dikatakan sebagai suatu subkelas perangkat lunak yang komputer manfaatkan kemampuanya secara langsung untuk melakukan suatu tugas yang diinginkan pengguna. (Hadaryudi, 2010).

Data adalah informasi yang telah diterjemahkan ke dalam bentuk yang lebih sederhana untuk melakukan suatu proses. Sehubungan dengan komputer saat ini dan media transmisi, data adalah informasi yang diubah menjadi bentuk digital biner. (Wahyudi, 2008).

File merupakan informasi yang disimpan pada system penyimpanan data massal dalam bentuk unit-unit data berukuran besar. Sebuah file yang tipikal dapat berisis dokumen teks lengkap, sebuah foto, atau sebuah video. (Sommerville, 2011).

kriptografi berarti penyelidikan tentang kode rahasia atau teknik yang mengubah data menjadi berbeda dari aslinya dengan menggunakan algoritma matematika sehingga orang yang tidak mengetahui kuncinya tidak akan dapat membongkar data tersebut. (KBBI).

Algoritma Simetri sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsinya. (Dony, 2008).

3.2 Algoritma *Blowfish*

Blowfish merupakan sebuah algoritma kunci simetri yang dirancang pada tahun 1993 oleh Bruce Schneier untuk menggantikan DES. *Blowfish* merupakan algoritma kriptografi kunci simetri dengan panjang blok 64. Ukuran kunci maksimal *Blowfish* adalah 56 karakter atau 448 bit.

Algoritma utama dari blowfish dibagi menjadi dua (2) subalgoritma yaitu bagian ekspansi kunci dan enkripsi-dekripsi data.

1. Ekspansi Kunci
   1) Inisialisaikan s-box dan p-box, s-box terdiri dari 4 buah 32-bit string dengan panjang 256 sedangkan p-box terdiri dari 18 buah 32-bit string. Nilai dari p-box dan s-box didapatkan dari bilangan hexa dari phi.
   2) Xor-kan P_1 dengan 32-bit awal kunci, xor-kan P_2 dengan 32-bit berikutnya dari kunci, hingga seluruh panjang kunci telah ter-xor-kan.
   3) Enkrip *all-zero string* menggunakan cara seperti langkah 1 dan 2
   4) Gantikan P_1 dan P_2 dengan hasil dari langkah 3.
   5) Enkripsikan keluaran langkah 3 dengan langkah 2 dengan *subkey* yang telah diubah.
   6) Gantikan P_3 dan P_4 dengan hasil dari langkah 5.
   7) Lakukan seterusnya hingga seluruh p-box teracak sempurna.
2. Enkripsi dan Dekripsi
   1) Masukan dari proses ini adalah 64 bit yang diinisialkan "x"
   2) Bagi x menjadi 2 buah bagian sama besar, xL (x kiri) sepanjang 32 bit, dan xR (x kanan) sepanjang 32 bit.
   3) Lakukan iterasi i = 1 hingga i = 16:
      xL = xL xor p[1]; (1)
      xR = F(xL) xor xR; (2)
      Tukar (xL, xR);
   4) Fungsi F adalah sebagai berikut: bagi xL menjadi 4 buah 8 bit a, b, c, dan d.
      $F(xl) = ((s[1,a] + s[2,b] \bmod 2^{32}) \text{ xor } s[3,c]) + s[4,d] \bmod 2^{32}$ (3)
   5) Langkah terakhir adalah:
      Swap (xL,xR);
      xR = xR xor p[17]; (4)
      xL = xLxor p[18]; (5)
      Gabungkan xL dan xR menjadi 64 bit *return* hasil gabungan.
   6) Pada proses dekripsi langkah-langkahnya sama persis dengan proses enkirpsi, hanya saja p-box digunakan dengan urutan terbalik.

3.3 *Tiny Encryption Algorithm* (TEA)

TEA merupakan suatu algoritma kunci simetri yang diciptakan oleh David Wheeler dan Roger Needham dari *Computer Laboratory*, Cambridge *University*, *England* pada bulan November 1994.

Sistem penyandian TEA menggunakan proses *feistel network* dengan menambah fungsi matematika 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 bit kunci dan data bercampur secara berulang-ulang.

Langkah enkripsi dekripsi TEA :

Delta = $(9E3779B9_{18})$
sum = sum + delta
y = y+(((z<<4)+k[0]) xor (z+sum) xor ((z>>5)+k[1]))
z = z+(((y<<4)+k[2])xor(y+sum)xor ((y>>5)+k[3]))

keterangan :
y = bagian kiri *plaintext*
z = bagian kanan *plaintext*
<< = geser bit kekanan
>> = geser bit kekiri
k[i] = bagian kunci ke-i

3.4 Metode *Rapid Application Development*

Rapid Aplication Development (RAD**)** adalah sebuah model proses perkembangan perangkat lunak *skuensial linear* yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi "kecepatan tinggi" dari model *skuensial linear* dimana perkembangan cepat dicapai dengan menggunakan model pendekatan konstruski berbasis komponen. (Kendall & Kendall 2010).

3.5 Desain

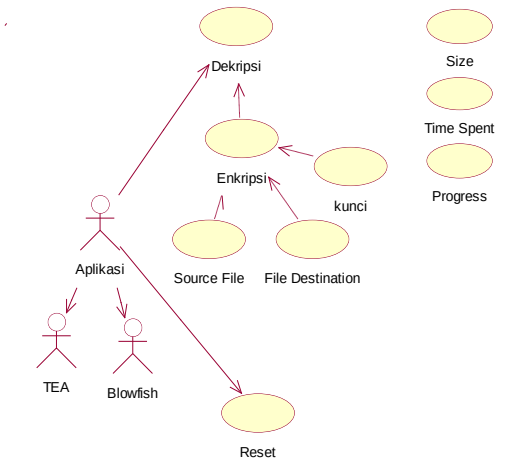
Dalam tahapan desain menggunakan UML

1. *Usecase* Global

Dari gambar *usecase* diatas dapat diketahui prosedur rancangan dan interaksi yang terdapat di dalam aplikasi. aplikasi terdiri dari dua class yaitu TEA dan *Blowfish* dimana masing-masing *class* memiliki *usecase* yang terdiri dari Dekripsi, Enkripsi, *Source File*, *File Destination* dan Kunci.

Dekripsi hanya dapat terjadi apabila syarat-syarat dekripsi telah terpenuhi, syarat dilakukan nya dekripsi adalah adanya *source file* yang telah dienkripsi, *file destination* dan kunci. Sedangkan agar Enkripsi dapat terjadi syarat yang harus dipenuhi adalah adanya *source file* yaitu *file* yang akan dienkripsi, *file destination* yaitu akan disimpan dengan nama apa dan folder mana *file* hasil enkripsi akan disimpan dan kunci. Untuk Reset digunakan untuk mengosongkan *source file*, *file destination* dan kunci.

Usecase Size, *Time Spent* dan *progress* merupakan *usecase* pendukung dari sistem yang berguna untuk menampilkan ukuran *file* dan waktu yang dihabiskan.

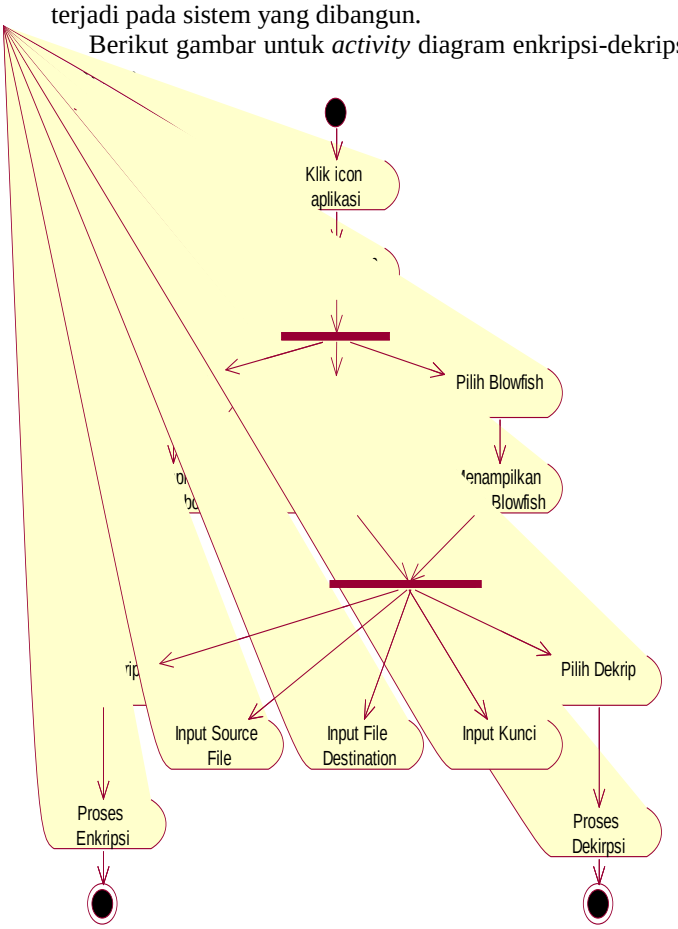


Gambar 1 *Usecase* Global

2. *Activity* Diagram enkripsi-dekripsi sistem

Pada *activity* diagram enkripsi dan dekripsi sistem, menggambarkan kejadian atau aktivitas-aktivitas yang terjadi pada sistem yang dibangun.

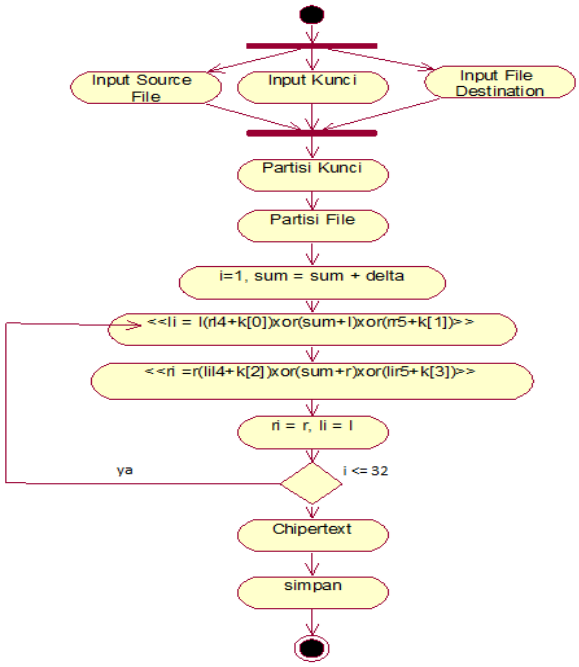
Berikut gambar untuk *activity* diagram enkripsi-dekripsi



Gambar 2 *Activity* diagram enkripsi-dekripsi sistem

3. *Activity* Diagram Enkripsi TEA

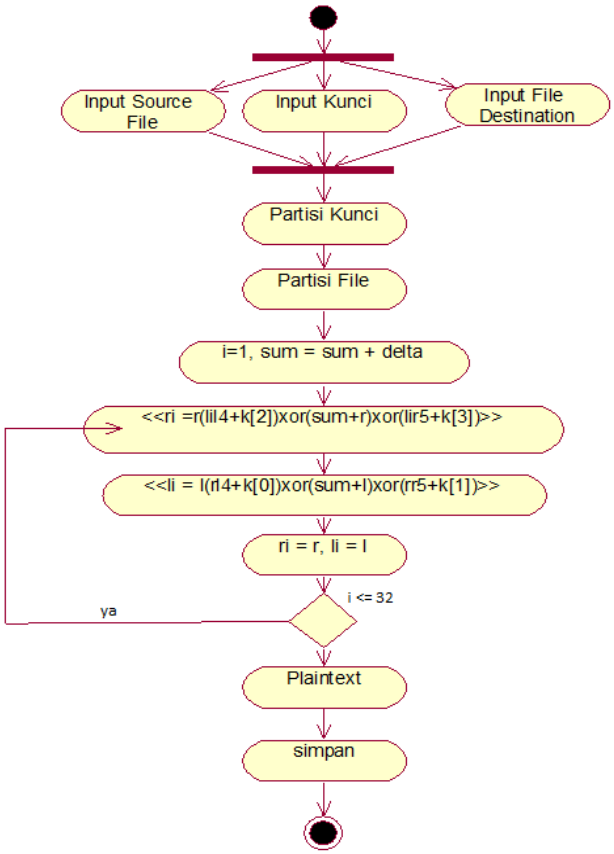
Pada diagram aktivitas enkripsi TEA, menggambarkan kejadian atau proses saat dilakukan proses enkripsi TEA. Dimana setelah Inputan dimasukkan maka akan dilakukan partisi kunci, dan partisi *file* dimana hasil partisi akan digunakan pada perhitungan algoritma TEA dan menghasilkan chipertext kemudian akan disimpan sesuat dengan *file destination* yang telah diinputkan.



Gambar 3 *Activity* Diagram Enkripsi TEA

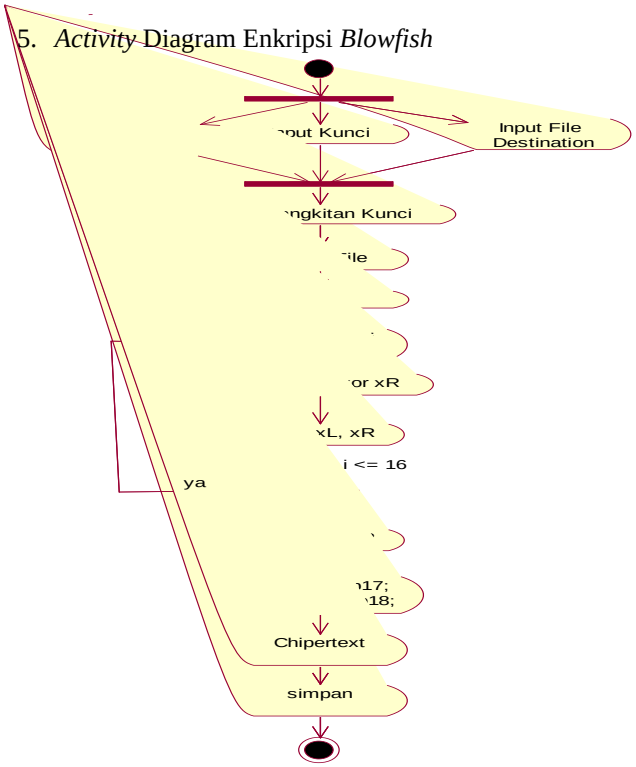
4. *Activity* Diagram dekripsi TEA

Pada diagram aktivitas dekripsi TEA, menggambarkan kejadian atau proses saat dilakukan proses dekripsi TEA. Pada dasarnya dekripsi memiliki proses yang sama dengan proses enkripsi, dimana setelah inputan dimasukkan maka akan dilakukan partisi kunci, dan partisi *file* dimana hasil partisi akan digunakan pada perhitungan algoritma TEA dan menghasilkan *plaintext* kemudian akan disimpan sesuat dengan *file destination* yang telah diinputkan.Berikut alur aktivitas dari dekripsi TEA :



Gambar 4 *activity* diagram dekripsi TEA

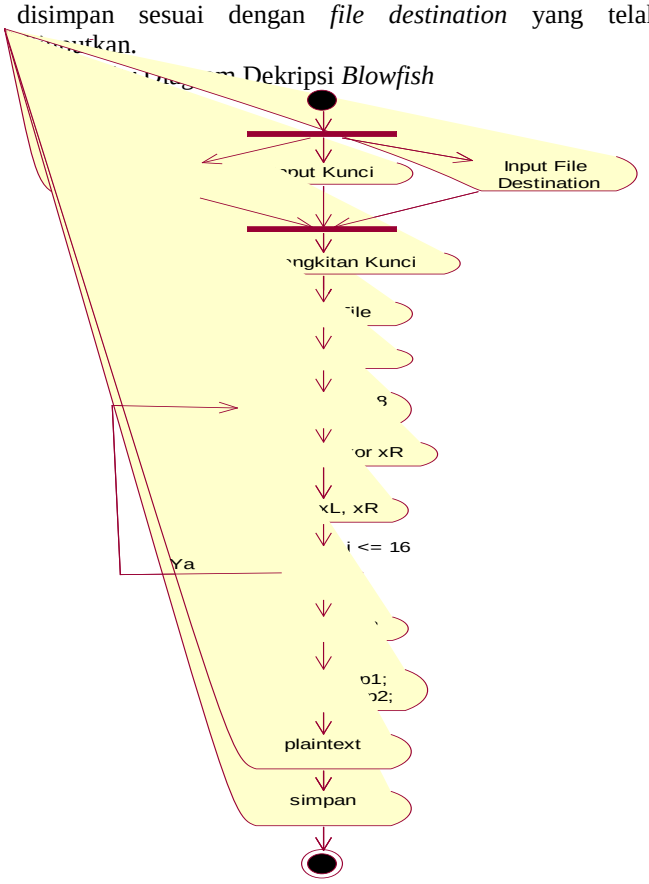
5. *Activity* Diagram Enkripsi *Blowfish*



Gambar 5 *activity* diagram enkripsi *Blowfish*

Gambar di atas merupakan diagram aktivitas saat proses enkripsi *Blowfish* berjalan. Sama hal-nya dengan proses enkripsi yang terjadi pada proses enkripsi tea inputan yang dimasukkan akan dilakukan partisi kemudian digunakan untuk melakukan proses enkripsi. *File* yang dienkripsi akan disimpan sesuai dengan *file destination* yang telah

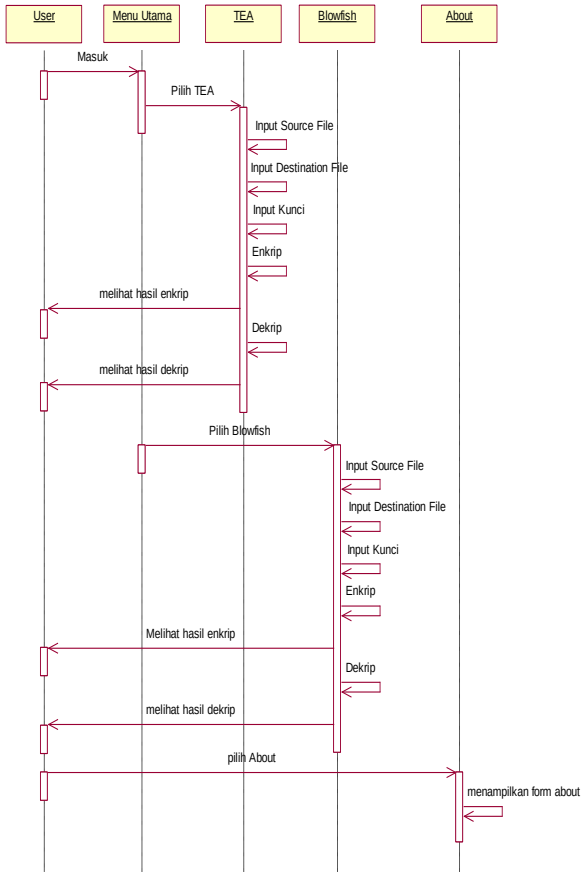
Dekripsi *Blowfish*



Gambar 6 *Activity* Diagram Dekripsi *Blowfish*

Gambar diatas merupakan diagram aktivitas saat proses dekripsi *Blowfish* berjalan. Sama hal-nya dengan proses dekripsi yang terjadi pada proses enkripsi *Blowfish* inputan yang dimasukkan akan dilakukan partisi kemudian digunakan untuk melakukan proses enkripsi. *File* yang dienkripsi akan disimpan sesuai dengan *file destination* yang telah diinputkan. Yang membedakan dengan proses enkripsi *Blowfish* adalah p-array atau p-box yang digunakan. Dimana pada dekripsi p-box digunakan secara terbalik mulai dari P_{18}.

7. *Sequence* Diagram



Gambar 7 *Sequence* Diagram

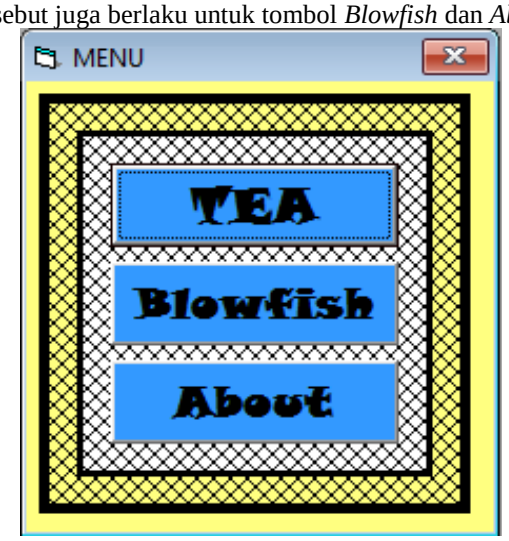
Diagram sekuensial digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *usecase*. Pada gambar diatas telihat bahwa saat *user* pertama masuk dalam aplikasi maka akan ditampilakan menu utama, saat *user* memilih TEA maka *form* Tea akan tampil. Dalam *form* tea, user dapat menginputkan *source file*, *destination file* dan kunci kemudian user dapat memilih antara enkrip atau dekrip, dan hasilnya dapat dilihat di folder yang telah diinputkan pada *destination file*.

3.6 Implementasi

1. Menu Utama

Form Menu utama memiliki 3 tombol pilihan yaitu TEA, *Blowfish*, dan *About* dimana kedua menu tersebut

digunakan untuk memasuki *form* dari menu yang dipilih. Apabila memilih tombol TEA maka *form* TEA akan tampil, hal tersebut juga berlaku untuk tombol *Blowfish* dan *About*.



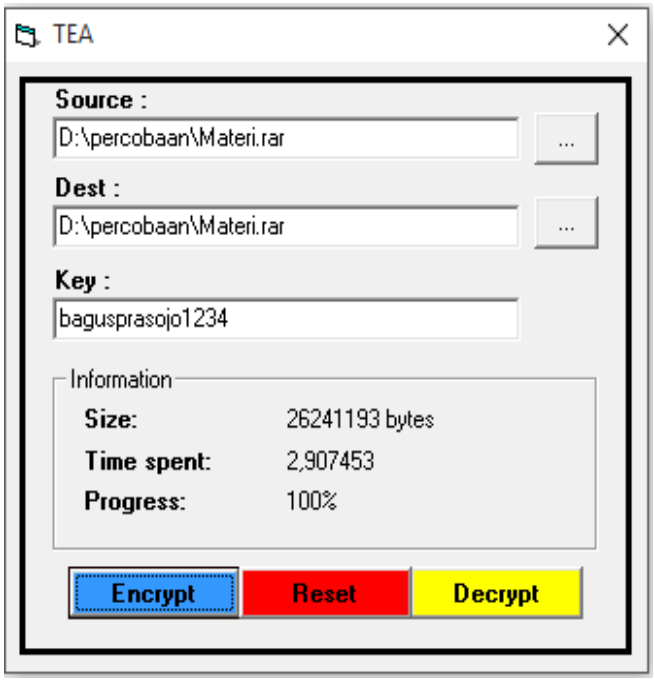
Gambar 8. Tampilan Menu Utama

2. Tampilan *Form* TEA

Dalam *form* ini berisikan 3 *textbox* yang harus diisi, *textbox source* digunakan untuk menginputkan *file* yang akan diproses. *Texbox* dest digunakan untuk menginputkan tempat penyimpanan dan nama *file* yang dihasilkan dari proses enkripsi ataupun dekripsi sedangkan *textbox key* digunakan untuk menginputkan kunci yang akan digunakan untuk melakukan proses enkripsi dan dekripsi. Pada *form* ini terdapat 5 *button* yaitu 2 *button search*, 1 *button encrypt*, 1 *button Reset*, 1 *button Dectypt* dengan fungsi sebagai berikut:

Button search digunakan untuk membuka fungsi *browse file* pada *windows*, *button encrypt* untuk memulai proses enkripsi, *button Reset* untuk membersihkan atau menghapus isian dari *textbox* dan *button Decrypt* digunakan untuk memulai proses dekripsi.

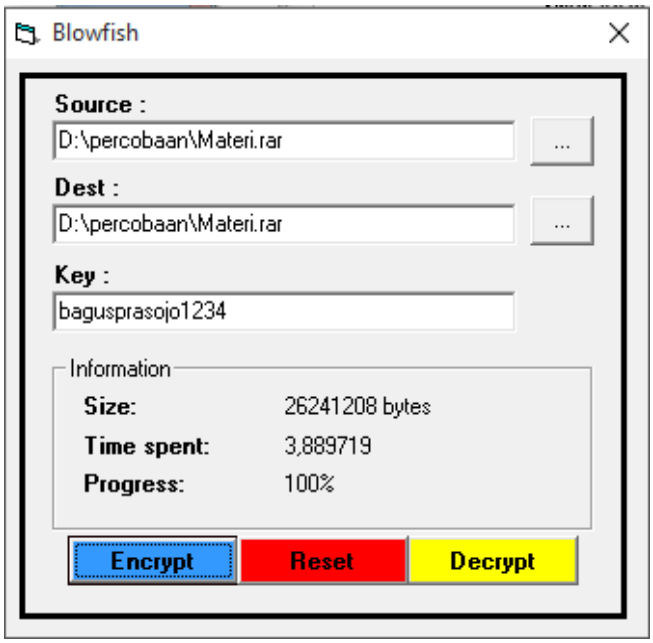
Berikut gambar untuk tampilan form TEA:



Gambar 9. Tampilan *Form* TEA

3. Tampilan *Form Blowfish*

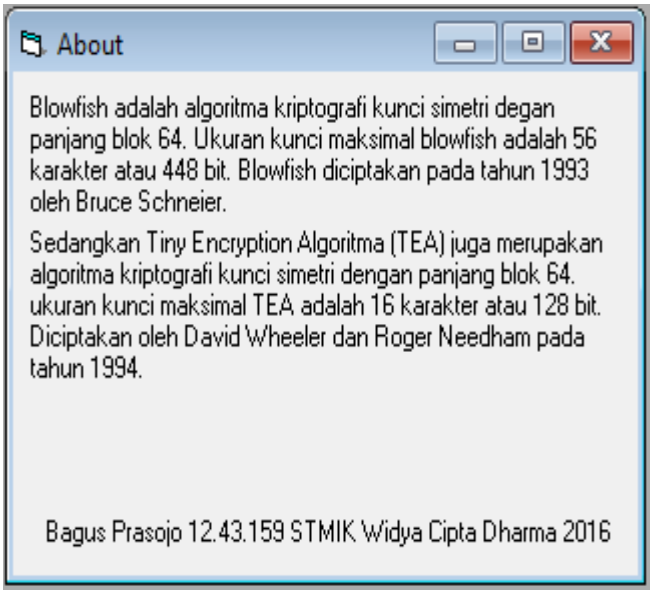
Form ini memiliki fungsi yang sama dengan antarmuka TEA. Hanya saja pada *form* ini proses enkripsi dan dekripsi menggunakan algoritma *Blowfish*.



Gambar. 10. Tampilan *Form Blowfish*

4. Tampilan *Form About*

Dalam *form About* berisikan informasi tentang sekilas tentang algoritma *Blowfish* dan TEA.



Gambar 11. Tampilan *Form About*

4. Hasil Penelitian

4. 1 Perbandingan Waktu Proses Enkripsi

Tabel 1 Pebandingan Waktu Proses Enkripsi

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.368	0,34	0,26
2	Appsurat.ppt	3.643.392	0,59	0,39
3	Img1.jpg	6.323.371	1,07	0,62
4	Bohemian.mp3	5.645.373	0,98	0,53
5	Materi.rar	26.241.193	3,93	2,45
jumlah		43.959.697	6,72	4,25
rerata		8.791.939	1,34	0,85

Dari lima (5) sempel yang digunakan untuk pengujian maka dapat diketahui jika dalam TEA waktu yang dibutuhkan untuk melakukan proses enkripsi lebih cepat dibandingakan dengan *Blowfish*.

4. 2 perbandingan Waktu Proses Dekripsi

Tabel. 2 Perbandingan Waktu Proses Dekripsi

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.384	0,37	0,24
2	Appsurat.ppt	3.643.408	0,62	0,42
3	Img1.jpg	6.323.384	0,99	0,68
4	Bohemian.mp3	5.645.392	0,90	0,59
5	Materi.rar	26.241.208	4,23	3,43
jumlah		43.959.697	7,11	5,36
rerata		8.791.939	1,42	1,07

Untuk mengetahui waktu proses dekripsi dari algotirma mana yang paling cepat, dilakukan perhitungan rerata dari lima (5) sample dengan hasil algoritma TEA memiliki waktu dekripsi lebih cepat dibandingkan dengan *Blowfish.*

4.3 Perbandingan Kecepatan Proses Enkripsi

Tabel. 3 Perbandingan Kecepatan Proses Enkripsi Pengujian ke-1

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.368	0,34	0,26
2	Appsurat.ppt	3.643.392	0,59	0,39
3	Img1.jpg	6.323.371	1,07	0,62
4	Bohemian.mp3	5.645.373	0,98	0,53
5	Materi.rar	26.241.193	3,93	2,45
jumlah		43.959.697	6,72	4,25
rerata		8.791.939	1,34	0,85
Kecepatan (B/s)			6.561.148	10.343.457

Tabel. 4 Perbandingan Kecepatan Proses Enkripsi Pengujian ke-2

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.368	0,34	0,23
2	Appsurat.ppt	3.643.392	0,56	0,35
3	Img1.jpg	6.323.371	0,95	0,59
4	Bohemian.mp3	5.645.373	0,97	0,54
5	Materi.rar	26.241.193	3,89	2,46
jumlah		43.959.697	6,71	4,17
rerata		8.791.939	1,34	0,83
Kecepatan (B/s)			6.561.148	10.592.698

Tabel 5 Perbandingan Kecepatan Proses Enkripsi Pengujian ke-3

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.368	0,39	0,26
2	Appsurat.ppt	3.643.392	0,61	0,35
3	Img1.jpg	6.323.371	0,98	0,62
4	Bohemian.mp3	5.645.373	0,87	0,53
5	Materi.rar	26.241.193	3,92	2,59
jumlah		43.959.697	6,77	4,35
rerata		8.791.939	1,35	0,87
Kecepatan (B/s)			6.512.547	10.105.677

Tabel 6 Perbandingan Kecepatan Proses Enkripsi Pengujian ke-1, 2 , dan 3

No	Hasil Pengujian	Kecepatan (B/s)		Perbandingan Hasil Pengujian
		Blowfish	**TEA**	
1	Ke-1	6.561.148	10.343.457	1 : 1,5765
2	Ke-2	6.561.148	10.592.698	1 : 1,6145
3	Ke-3	6.512.547	10.105.677	1 : 1,5517
rerata				1 : 1,5809

Dari tiga (3) Pengujian yang dilakukan, dapat disimpulkan bahwa *Blowfish* dan TEA memiliki nilai perbandingan 1 : 1,5809 dimana TEA memiliki kecepatan 1,5809 kali lebih cepat dibandingkan dengan *Blowfish*. Nilai perbandingan kecepatan didapatkan dari rerata kecepatan dari tiga (3) percobaan yang dilakukan dengan menggunakan spesifikasi *Software* dan *Hardware* seperti yang diterangkan di bab sebelumnya. Percobaan kecepatan memiliki hasil yang dapat berubah-ubah dikarenakan beberapa faktor yang dapat mempengaruhi kinerja dari aplikasi, seperti *memory* yang tersedia dan kecepatan *processor*.

4.4 **Perbandingan Kecepatan Proses Dekripsi**

Tabel 7 Perbandingan Kecepatan Proses Dekripsi Pengujian ke-1

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.384	0,38	0,34
2	Appsurat.ppt	3.643.408	0,60	0,43
3	Img1.jpg	6.323.384	1,04	0,64
4	Bohemian.mp3	5.645.392	0,90	0,59
5	Materi.rar	26.241.208	4,24	2,71
jumlah		43.959.776	7,16	4,71
rerata		8.791.955	1,43	0,94
Kecepatan (B/s)			6.063.417	9.254.689

Tabel 9 Perbandingan Kecepatan Proses Dekripsi Pengujian ke-3

Tabel 10 Perbandingan Kecepatan Proses Dekripsi Pengujian ke-1, 2, dan 3

No	Nama	Ukuran (byte)	Waktu (s)	
			Blowfish	**TEA**
1	Bab IV.docx	2.106.384	0,37	0,25
2	Appsurat.ppt	3.643.408	0,64	0,40
3	Img1.jpg	6.323.384	1,01	0,65
4	Bohemian.mp3	5.645.392	0,98	0,57
5	Materi.rar	26.241.208	4,28	2,90
jumlah		43.959.776	7,28	4,77
rerata		8.791.955	1,45	0,95
Kecepatan (B/s)			6.063.417	9.254.689

No	Nama	Ukuran (byte)	Waktu (s)		Kecepatan (B/s)		Perbandingan Hasil Pengujian
			No	Hasil Pengujian	***Blowfish***	**TEA**	
1	Bab IV.docx	2.106.384	1	0,37 Ke-1	6.191.517	8.216.780	1 : 1,3271
2	Appsurat.ppt	3.643.408	2	0,62 Ke-2	6.063.417	9.254.689	1 : 1,5263
3	Img1.jpg	6.323.384	3	0,99 Ke-3	6.063.417	9.254.689	1 : 1,5263
4	Bohemian.mp3	5.645.392		0,90 **rerata**	0,59		1 : 1,4839
5	Materi.rar	26.241.208		4,23	3,43		
jumlah		43.959.776		7,11	5,36		
rerata		8.791.955		1,42	1,07		
Kecepatan (B/s)				6.191.517	8,216,780		

Tabel 8Perbandingan Kecepatan Proses Dekripsi Pengujian ke-2

Dari tiga (3) pengujian yang dilakukan untuk mengetahui kecepatan proses dekripsi diatas dihasilkan nilai perbandingan antara *Blowfish* dan TEA adalah 1 : 1,4879 dimana TEA memiliki kecepatan yang lebih tinggi dibandingkan *Blowfish* dalam hal proses dekripsi.

4.5 Pengujian Keamanan

Tabel 11 Pengujian Keamanan

No	Nama	keterangan	Kesimpulan

		Blowfish	**TEA**	
1	*iSunShare*	Tidak Berhasil	Tidak Berhasil	Aman
2	*Eassy office recovery*	Tidak Berhasil	Tidak Berhasil	Aman
3	*S2 recovery tool*	Tidak Berhasil	Tidak Berhasil	Aman
4	*Savvy docx recovery*	Tidak Berhasil	Tidak Berhasil	Aman
5	*Accent office recovery*	Tidak Berhasil	Tidak Berhasil	Aman

Pengujian keamanan ini dilakukan menggunakan beberapa *software recovery* yang biasa digunakan untuk me-*recovery password* ataupun *file*. Dengan kesimpulan *software recovery* tidak dapat mengembalikan *file* yang telah dienkrip menjadi *file* semula dan tidak dapat mengetahui kunci yang digunakan untuk proses enkripsi sehingga dapat dikatakan proses enkripsi berjalan sesuai tujuan dan aman.

5. KESIMPULAN & SARAN

5.1 Kesimpulan

Berdasarkan dengan hasil penelitian yang dilakukan dan uraian-uraian yang dibahas dalam bab-bab sebelumnya, maka dapat ditarik kesimpulan bahwa:

1. *Tiny Encryption Algorithm* (TEA) memiliki performa yang jauh lebih cepat dibandingkan dengan *Blowfish.*
2. Aplikasi ini dapat menjadi sarana perbandingan antara *Blowfish* dan TEA dalam hal waktu yang dibutuhkan untuk melakukan enkripsi dan dekripsi.
3. Hasil dari perbandingan implementasi kedua algoritma menunjukkan bahwa TEA lebih unggul dibandingkan *Blowfish* sebesar 1 : 1,5809 untuk kecepatan enkripsi, dan 1 : 1,4839 untuk kecepatan dekripsi. Dengan demikian, berdasarkan kecepatan, di mana kecepatan bergantung pada waktu proses, maka TEA lebih unggul dibanding *Blowfish*. Sedangkan dari segi keamananan, keduanya aman menurut *recovery software*.
4. Aplikasi ini dapat menjadi alternatif lain sebagai pengaman file atau data.

5.2 Saran

Adapun saran-saran dari penulis sebagai berikut:

1. Aplikasi ini masih sangat sederhana dan dapat ditingkatan menjadi aplikasi yang lebih baik lagi.
2. Aplikasi ini hanya dapat melakukan enkripsi dan dekripsi terhadap file secara satu persatu, diharapkan dalam penelitian selanjutnya dapat dikembangkan agar dapat mengenkripsi beberapa file secara bersamaan.
3. Dalam penelitian ini dilakukan perbandingan antara *Blowfish* dan TEA, diharapkan pada penelitian selanjutnya dapat melakukan perbandingan antara *Blowfish* ataupun TEA dengan algoritma *cryptografi* lainnya.

6. DAFTAR ISI

Ariyus, Dony, 2008, *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi,* Yogyakarta: Penerbit Andi

Dhanta, Risky, 2009, *Kamus Istilah Komputer Grafis dan Internet*, surabaya : Indah.

Galih Wahyu Prasetyo, 2010, *Aplikasi Enkripsi SMS Menggunakan Metode Blowfish*, Program Studi Teknik Informatika, Surabaya :Politeknik Elektronika Negeri Surabaya

http://www.cix.co.uk/~klockstone/tea, diakses tanggal 10 maret 2016

http://www.geocities.com/herong_yang/crypto/chiper_blowfish.html, diakses tanggal 10 maret 2016

http://www.schneier.com/blowfish.html, diakses tanggal 10 Maret 2016

Kemendikbud, 2008, *Kamus Besar Bahasa Indonesia Edisi ke-empat*,

Nogroho A , 2010, *Rekayasa Perangkat Lunak UML dan JAVA*, Yogyakarta : Andi Offset

Pressman, Roger S, 2010, *Software Engineering Practitioner's Approach Seventh Edition*, New York: The McGraw-Hill Companies, Inc

Rosa A.S dan M. Shalahuddin, 2011, *Modul Pembelajaran: Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Modula

Rosa A.S dan M. Shalahuddin, 2015, *Rekayasa Perangkat Lunak Cetakan Ketiga*, Bandung : Informatika Bandung

Sari, Yunita. 2009 *Perancangan dan Implementasi Sistem Keamanan Data Menggunakan Algoritma Simetri TEA,* Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Medan :Universitas Sumatera utara.

Sommerville, Ian. 2011, *Software Engineering Ninth Edition* .Adison – Wesley.

Vikram Reddy Andem. 2003, *A Cryptanalysis of The Tiny Encryption Algorithm*, Universitas of Alabama

William, Khandar. 2009, *Studi Mengenai Tiny Encryption Algorithm (TEA) dan Turunan- Turunannya*. Program Studi Teknik Informatika, Bandung : Institut Teknologi Bandung