IMPLEMENTASI ALGORITMA SEAL PADA KEAMANAN DATA

I Made Kartika, Restyandito, Sri Suwarno
Fakultas Teknologi Informasi, Program Studi Teknik Informatika
Universitas Kristen Duta Wacana Yogyakarta
Email: 22043639@students.ukdw.ac.id, ditto@ukdw.ac.id, sswn@ukdw.ac.id

Abstrak :

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu pesan, data, atau informasi. Hal ini sangat terkait dengan betapa pentingnya pesan, data, atau informasi tersebut dikirim dan diterima oleh pihak atau orang yang berkepentingan, apakah pesan, data, atau informasi masih *authentic*. Pesan, data, atau informasi akan tidak berguna lagi apabila di tengah jalan informasi itu disadap atau dibajak oleh orang yang tidak berhak atau berkepentingan. Oleh karena itu dibutuhkan suatu siatem yang dapat membantu dalam menjaga keamanan maupun kerahasiaan suatu data.

Kriptografi merupakan salah satu metode pengamanan data yang dapat digunakan untuk menjaga keamanan maupun kerahasiaan dari suatu data. Dari permasalah ini penulis akan menimplementasikan algoritma kriptografi SEAL dalam menjaga kerahasiaan dan keamanan data dengan melalui proses enkripsi dan deskripsi, sehingga orang yang tidak berhak tidak dapat membaca informasi yang ada didalam data tersebut.

Kata Kunci : *Kriptografi, Enkripsi, Stream Chiper, SEAL*

1. Pendahuluan

Arus informasi yang semakin marak untuk menggunakan media digital, dalam bentuk file memungkinkan orang lain untuk bisa membuka data yang bukan haknya. Rahasia perusahaan, data-data penting menjadi sangat berarti untuk dilakukan proses enkripsi. Hal ini sangat membantu bagi pihak-pihak yang berkepentingan dengan data untuk melindungi datanya dari orang pihak-pihak lain..

Pada kenyataannya ada banyak sekali tipe data yang dapat di-enkripsi, salah satu diantaranya adalah data teks. Saat ini sudah ada beberapa algoritma untuk menjaga keamanan data teks, seperti DES, AES (Rijndael), Blowfish, TEA, XOR256 Block, XOR256 Stream dan SEAL. Masing-masing algoritma memiliki kelebihan yang berbeda-beda. Dalam penelitian ini

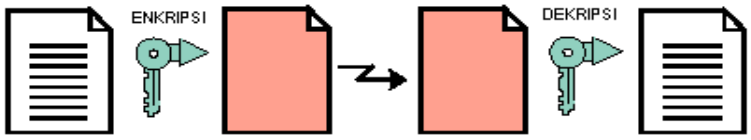
penulis akan membahas mengenai algoritma SEAL untuk pengamanan data file teks.

Algoritma Seal merupakan salah satu algoritma kriptografi untuk mengamankan data. Oleh sebab itu, perlu dibangun sebuah sistem untuk mengaplikasikan algoritma SEAL dengan melibatkan proses enkripsi dan dekripsi sehingga dapat dianalisis lebih dalam mengenai struktur *chiper* dan cara kerja algoritma kriptografi SEAL dengan beberapa pengujian sehingga dapat dilakukan analisa terhadap perubahan hasil keluaran *(output)* yang terjadi jika terjadi perubahan pada masukan *(input)* sehingga dapat diketahui seberapa baik proses enkripsi dengan algoritma kriptografi SEAL dalam memproteksi data.

2. Landasan Teori

2.1 Kriptografi

Secara etimologi (ilmu asal usul kata), kata kriptografi berasal dari gabungan dua kata dalam bahasa Yunani yaitu "*kriptos*" dan "*graphia*". Kata *kriptos* digunakan untuk mendeskripsikan sesuatu yang disembunyikan, rahasia atau misterius. Sedangkan kata *graphia* berarti tulisan. Kriptografi adalah seni dan ilmu dalam mengamankan pesan. Dalam dunia kriptografi, pesan disebut *plaintext* atau *cleartext*. Proses untuk menyamarkan pesan dengan cara sedemikian rupa untuk menyembunyikan isi aslinya disebut enkripsi. Pesan yang telah dienkripsi disebut *ciphertext*. Proses pengembalian sebuah *ciphertext* ke *plaintext* disebut dekripsi.



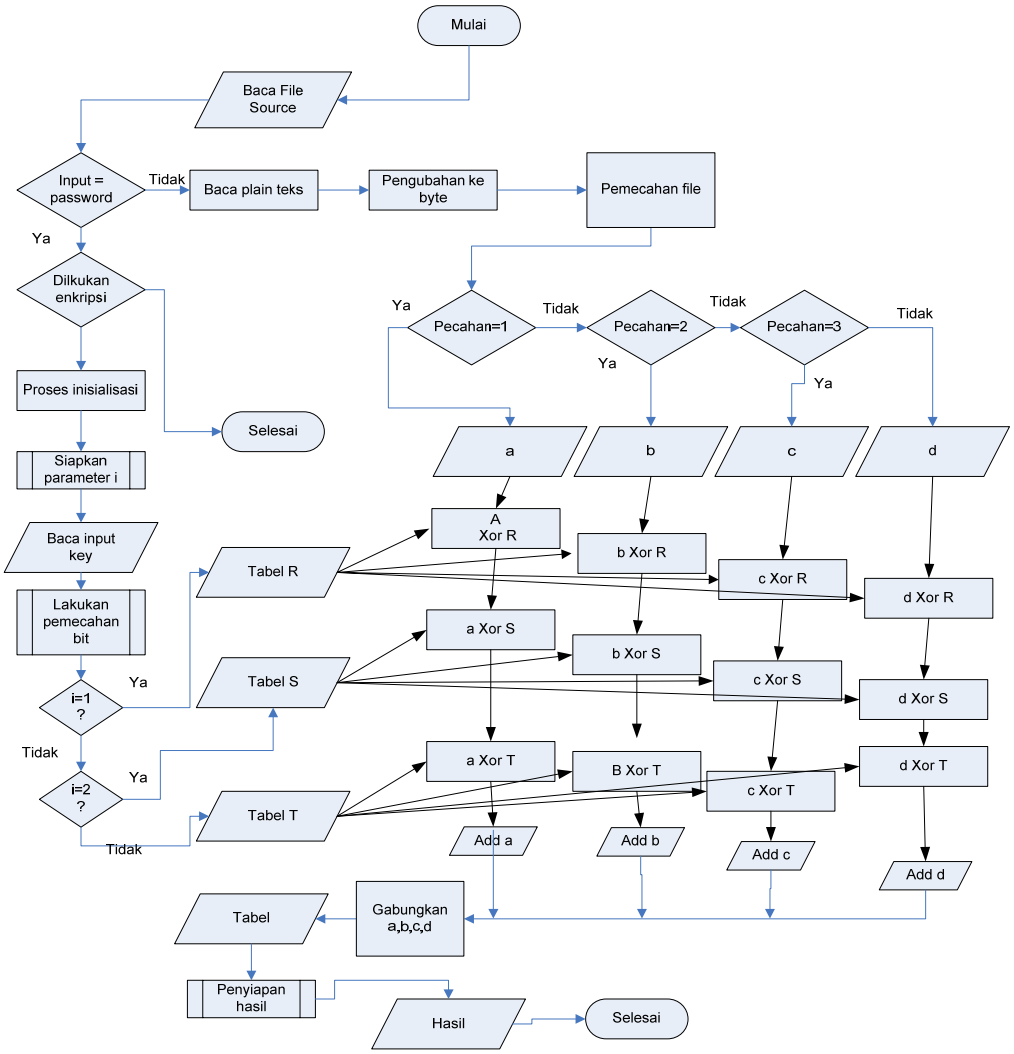
Gambar 1. Konsep Dasar dari Enkripsi dan Dekripsi

2.2 *Stream Chiper*

Stream cipher adalah suatu algoritma simetrik enkripsi yang sangat penting. Algoritma ini bekerja dengan cara mengenkrip karakter demi karakter (biasanya digit *binary*) dari *plaintext*. *Stream cipher* jauh lebih cepat dibandingkan dengan algoritma *block cipher*. Algoritma ini secara umum digunakan untuk mengenkrip *plaintext* yang kecil biasanya dalam ukuran *bit.* Suatu *stream cipher* menghasilkan apa yang disebut suatu *keystream* (suatu barisan *bit* yang digunakan sebagai kunci) untuk menghasilkan *pseudo-random sequence* yang diinisialisasikan dengan menggunakan kunci rahasia yang kemudian dilakukan operasi XOR dengan *plaintext* untuk menghasilkan *ciphertext*.

2.3 Algoritma SEAL

SEAL merupakan singkatan dari "*Software Encryption Algorithm*", yang merupakan salah satu algoritma enkripsi *stream chiper* yang dibuat oleh Rogaway dan Coppersmith dan dipatenkan oleh perusahaan IBM pada tahun 1993. SEAL digunakan secara luas pada beberapa aplikasi dan umumnya dinyatakan sangat aman. Sampai saat ini belum diketahui siapa yang dapat memecahkan/membongkarnya.

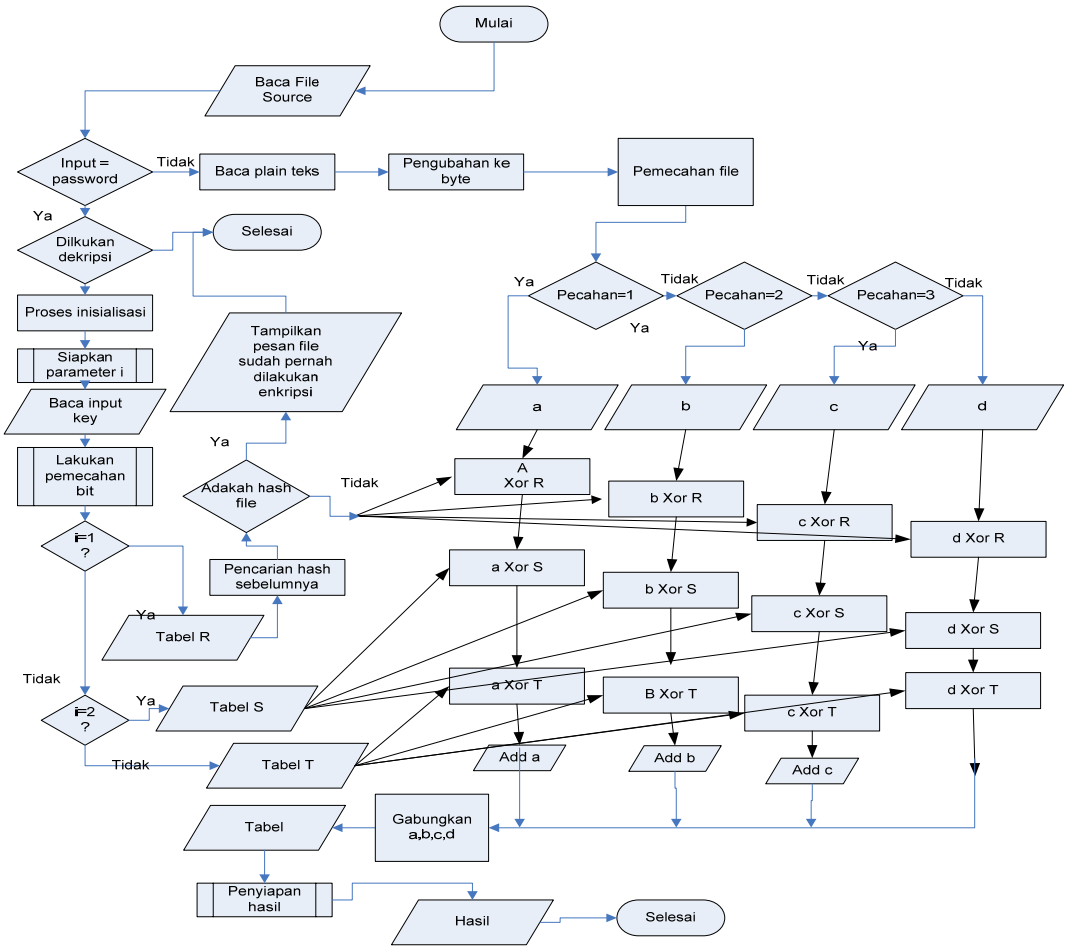


Gambar 2. Diagram Blok proses enkripsi dalam metode SEAL

a. Cara Kerja Enkripsi SEAL

Untuk proses enkripsi komputer mengambil data dari file *ciphertext* dan password. *Ciphertext* dan password diubah ke dalam bentuk *byte*. Password yang berupa kumpulan *byte* merupakan input untuk diproses ke dalam algoritma SHA (*Secure Hash Algorithm*). Hasil dari SHA kemudian dimasukkan kembali ke dalam algoritma SHA yang telah

dimodifikasi sehingga terbentuk tabel T, S, R. Pada iterasi pertama *ciphertext byte* dilakukan operasi XOR dengan tabel S, sehingga diperoleh *output* a,b,c,d. Kemudian masing-masing *output* tersebut dilakukan proses XOR dengan tabel T, dan yang terakhir dilakukan operasi XOR dengan tabel R. Setelah iterasi terakhir *output* a,b,c,d ditambahkan menjadi satu dan hasil penambahan tersebut disimpan ke dalam *output.* Gambar 2 menunjukkan alur kerja proses enkripsi dengan algoritma SEAL.



Gambar 3. Diagram Blok Proses dekripsi dalam metode SEAL

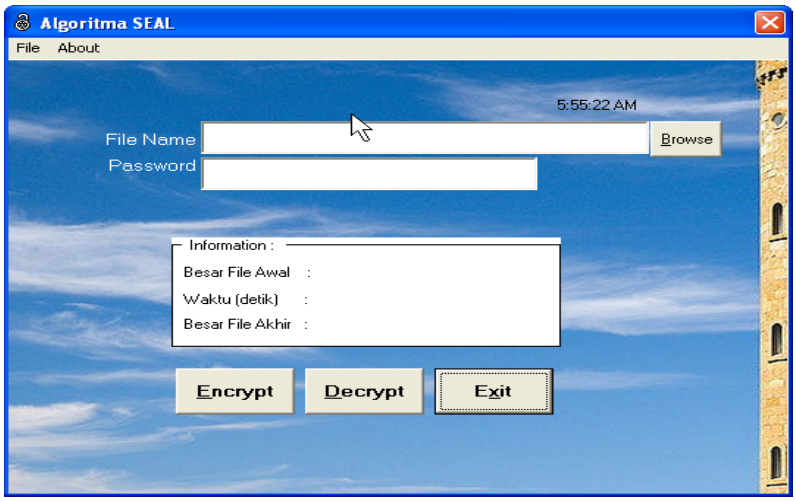
b. Cara Kerja Deskripsi SEAL

Untuk proses deskripsi SEAL pada umunya hampir sama pada proses enkripsi SEAL. Komputer mengambil data dari file *ciphertext* dan *password*. *Ciphertext* dan *password* diubah ke dalam bentuk *byte*. *Password* yang berupa kumpulan *byte* merupakan input untuk diproses ke dalam algoritma SHA (*Secure Hash Algorithm)*. Hasil dari SHA kemudian dimasukkan kembali ke dalam algoritma SHA yang telah dimodifikasi sehingga terbentuk tabel T, S, R. Pada iterasi pertama *ciphertext byte* dixorkan dengan tabel S,

diperoleh a,b,c,d. Kemudian masing-masing output tersebut dixorkan dengan tabel T, dan yang terakhir xor dengan tabel R. Setelah iterasi terakhir output a,b,c,d diadd menjadi satu dan hasil add tersebut disimpan ke dalam *output*. Gambar 3 menujukkan proses dekompresi algoritma SEAL.

3. Implementasi dan Pembahasan

Gambar 4. merupakan tampilan dari form utama dalam program yang dibuat. Yang digunakan untuk proses enkripsi maupun deskripsi .Tampilan utama ini dibuat sesederhana mungkin, namun dapat berfungsi sesuai dengan hal yang diharapkan, dalam tampilan menu utama tersebut terdapat beberapa bagian masukan dan tombol yang digunakan dalam proses enkripsi dan deskripsi.



Gambar 4. Tampiln Menu Utama Sistem

Seperti misalnya tempat untuk memasukkan data yang akan di proses baik yang akan dienkripsi maupun dideskripsi, ada juga inputan untuk mencatat password yang digunakan dimana password disini harus sama antara password yang digunakan untuk proses enkripsi dengan password yang digukana untuk proses deskripsi, kemudian ada 2 tombol yang digunakan untuk proses ekripsi dan deskripsi.

4. Analisis Hasil

4.1 Analisis ukuran file terhadap proses enkripsi / deskripsi

Tabel 1. Perbandingan besar ukuran file pada proses enkripsi dan dekripsi

Nama File	Besar ukuran file awal (byte)	Besar ukuran file setelah Enkripsi (byte)	Selisih ukuran file (byte)	Bessar perubahan ukuran file(%)
komp1.txt	23	88	65	282.60%
komp9.txt	1242	1304	62	4.99%
komp11.txt	2024	2088	64	3.16%
komp13.txt	3887	3952	65	1.67%
duta1.txt	5520	5584	64	1.15%
Sepeda.jpg	988	1046	58	5.87%
Gunung.jpg	1371	1431	60	4.37%
Baju.jpg	1677	1739	62	3.69%
Buku_tulis.jpg	1777	1842	65	3.65%
Palm.jpg	1803	1863	60	3.32%
Pengatar algoritma.doc	1042	1104	62	5.95%
Seal 2.doc	29696	29760	64	0.21%
Keamanan data.doc	47975	48037	62	0.12%
Onno_tcp_ip.doc	55808	55888	80	0.14%
Pengantar enkripsi.doc	69072	69156	84	0.12%
Enkripsi.pdf	2472	2488	16	0.64%
Pengatar multimedia.pdf	3573	3592	19	0.53%
Database.pdf	6357	6376	19	0.29%
Keamanan jaringan.pdf	24061	24080	19	0.07%
Rogaway_seal.pdf	33388	33408	20	0.05%

Dari tabel 1 di atas dapat dilihat bahwa setiap file mengalami perubahan ukuran besar file setelah dilakukan proses enkripsi.. Hal ini disebabkan oleh penggunaan proses pencarian tabel T, R, dan S, serta proses xor dari tabel T, R, dan S dengan data dalam bagian a,b,c dan pemecahan data file menjadi empat bagian, proses looping membuat perbedaan antar

kapasitas. Dari setiap file mengalami perubahan besar ukuran file dengan kisaran besar yang hampir sama sehingga tidak ada perlakuan khusus terhadap suatu file. Dengan perubahan ukuran besar file dengan kisaran yang hampir sama menyebabkan semakin besar ukuran file maka semakin kecil porsentase perubahan ukuran besar filenya.

4.2 Analisis waktu proses terhadap besar ukuran file

Tabel 2. Perbandingan waktu proses enkripsi dan deskripsi

Nama File	Besar ukuran file (Byte)	Waktu proses Enkripsi (detik)	Waktu proses Deskripsi (detik)	Selisih Waktu (detik)	Bessar perubahan waktu(%)
komp3.txt	138	0.09838	0.30200	0.20362	206.97%
komp9.txt	1242	0.12988	0.27000	0.14012	107.88%
komp11.txt	2024	0.08313	0.25488	0.17175	206.60%
komp13.txt	3887	0.11425	0.30138	0.18713	163.78%
duta1.txt	5520	0.12963	0.31763	0.188	145.02%
Sepeda.jpg	988	0.036375	0.037341	0.00096	2.65%
Gunung.jpg	1371	0.26365	0.27346	0.00981	3.72%
Baju.jpg	1677	0.26456	0.27386	0.0093	3.51%
Buku_tulis.jpg	1777	0.236146	0.26358	0.02743	11.61%
Palm.jpg	1803	0.236569	0.261576	0.025	10.57%
Pengatar algoritma.doc	1042	0.020937	0.022188	0.001251	5.87%
Seal 2.doc	29696	0.10781	0.12055	0.01274	11.81%
Keamanan data.doc	47975	0.164875	0.186094	0.021219	12.86%

Nama File	Besar ukuran file (Byte)	Waktu proses Enkripsi (detik)	Waktu proses Deskripsi (detik)	Selisih Waktu (detik)	Bessar perubahan waktu(%)
Onno_tcp_ip.doc	55808	0.21406	0.24063	0.02657	12.41%
Pengantar enkripsi.doc	69072	0.236031	0.257562	0.021531	9.12%
Enkripsi.pdf	2472	1.019344	1.18969	0.170346	16.71%
Pengatar multimedia.pdf	3573	1.02125	1.131094	0.109844	10.75%
Database.pdf	6357	1.021063	1.118875	0.097812	9.57%
Keamanan jaringan.pdf	24061	2.20594	2.34312	0.13718	6.21%
Rogaway_seal.pdf	33388	2.033687	2.119938	0.086251	4.24%

Waktu proses enkripsi dan deskripsi file tergantung pada ukuran file yang akan di enkripsi maupun di deskripsi. Dari beberapa percobaan, seperti yang ditunjukkan hasilnya pada Table 2, dapat dilihat waktu yang dibutuhkan setiap file untuk proses enkripsi maupun deskripsi tidak begitu berbeda jauh semakin besar ukuran file yang diproses maka waktu yang dibutuhkan untuk proses enkripsi dan deskripsi juga semakin lama. Hal ini terjadi karena semakin besar ukuran file maka makin lama waktu yang dibutuhkan sistem untuk memecah file maupun untuk melakukan proses perulangan. Semakin lama waktu yang dibutuh pada proses enkripsi maupun deskripsi maka besar perubahan prosentase waktunya semakin kecil dimana kisaran waktu yang dibutuhkan setiap file tidak jauh berbeda sehingga tipe file tidak mempengruhi waktu maupun ukuran file yang dienkripsi maupun yang dideskripsikan.

4.3 Analisis pada file dengan ukuran yang hampir sama

Tabel 3. Perbandingan antara Ukuran file *plaintext, gambar, dokumen dan pdf* dengan waktu enkripsi

No	Nama File	Besarnya file sebelum enkripsi	Besar file setelah enkripsi	Waktu Enkripsi	Waktu Dekripsi
1	enkripsi 1.txt	140	202	0.09838	0.11048
2	analisis 2.txt	200	262	0.08271	0.09901
3	komputer.doc	147	211	0.08000	0.09420
4	algoritma dasar.doc	195	261	0.08250	0.09480
5	robot atas.jpg	150	212	0.08105	0.09725
6	dinding.jpg	190	260	0.08263	0.09923
7	silabus.pdf	154	220	0.08300	0.09840
8	Perancangan sistem.pdf	193	263	0.08245	0.09875

Sebenarnya di analisis ini akan dijelaskan untuk perbandingan antar file teks, dokumen, gambar dan pdf dengan kapasitas awal sama, namun karena sangat sulit untuk mencari file dengan kriteria keempat-empatnya sama, maka hanya di berikan contoh untuk perbandingan dengan kapasitas yang tidak jauh berbeda diantara masing-masing file. Dari hasil tabel 3 tersebut terlihat bahwa untuk kapasitas awal dengan akhir berbeda, ini terjadi karena kapasitas yang diinginkan tidak tercapai kesamaannya, sedangkan untuk hasil akhir terdapat sedikit perbedaan, ini sebenaranya hampir sama dengan untuk masing-masing percobaan. Demikian juga untuk waktu yang diperlukan dalam proses enkripsinya. Dengan demikian untuk masing-masing contoh dari percobaan ini, dapat disimpulkan bahwa bentuk file tidak begitu berpengaruh. Hal ini disebabkan proses atau perlakuan yang terjadi pada proses enkripsi maupun dekripsi di masing-masing file adalah sama.

4.4 Analisis pada file dengan panjang password berbeda

Tabel 4. Perbandingan hasil enkripsi dengan panjang password yang berbeda

Nama File	Besar file awal (byte)	Panjng Password	Besar file setelah enkripsi (byte)	Panjang Password	Besar file setelah enkripsi (byte)
komp1.xt	23	1	88	4	88
komp2.txt	92	2	152	6	152
komp3.txt	138	4	200	8	200
komp4.txt	184	6	248	20	248
Clouds.jpg	1578	8	1640	4	1640
Baru.jpg	145	10	205	6	205
Mangga.jpg	313	1	377	8	377
Meja.jpg	518	2	581	10	581
Pengantar algoritma.doc	1042	4	1104	4	1104
Sql tingkat dasar.doc	1226	6	1286	6	1286
Belajar sql.doc	1637	8	1701	8	1701
Database 3.doc	1452	10	1515	15	1515
Pengantar multimedia.pdf	3573	1	3592	4	3592
Belajar sql.pdf	6134	2	6152	8	6152
Studi sql.pdf	8567	4	8584	10	8584
Database.pdf	6357	6	6376	14	6376

Dari hasil percobaan pada tabel 5 diatas dapat disimpulkan bahwa panjang password tidak berpengaruh terhadap ukuran besar file maupun waktu dari proses enkripsi maupun proses deskripsi.

5. Kesimpulan dan Saran

Dari hasil analisis pada program enkripsi dan dekripsi, dapat ditarik kesimpulan sebagai berikut :

1. Dalam implementasi program dengan menggunakan algoritma SEAL dapat dilakukan

proses enkripsi dan dekripsi pada file teks, gambar, dokumen word dan pdf .

2. Dalam proses enkripsi maupun deskripsi tipe file tidak begitu mempengaruhi hasil dari proses enkripsi dan deskripsi. Hasil besar ukuran file ataupun hasil kecepatan setelah proses enkripsi dan deskripsi tidak jauh berbeda dan rata rata hampir sama untuk setiap percobaan, hal ini dikarenakan dalam proses enkripsi maupun dekripsi file dibuka secara binary, sehingga tidak memandang bentuk tipe file. Dengan demikian ukuran file mempengaruhi waktu kecepatan secara simetris dalam proses enkripsi dan dekripsi.
3. Terjadinya perubahan besar ukuran file maupun waktu setelah proses enkripsi dikarenakan pada saat proses enkripsi file mengalami beberapa proses yaitu pembentukan hash file, penguraian data, penyisipan data dengan hash file, dan proses pembentukan kembali data file yang di enkripsi.
4. Panjang password pada waktu melakukan proses enkripsi maupun proses deskripsi tidak mempengaruhi besar ukuran file dan waktu yang dibutuhkan file pada waktu proses enkripsi dan deskripsi.

Daftar Pustaka

[1] Ariyus, D. (2006). Kriptografi keamanan data dan komunikasi. Penerbit Graha Ilmu Yogyakarta.

[2] Gilbert, H. (2003). Cryptanalisis of the SEAL encryption algorithm. France Telecom-CNET.

[3] Menezes, A., J., Oorschot, P. C. V., Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.

[4] Ramadhan, A. (2004). Microsoft visual basic 6.0, Elex Media Komputindo, Jakarta.

[5] Rogaway, P., Coppersmith, D. (1997). A software-optimized encryption algorithm. Cambridge Security Workshop, Springer-Verlag.

[6] Schneier, B. (1996). Applied cryptography, 2nd edition. John Wiley & Sons.

[7] Welschenbach, M. (2001). Cryptography in C and C++. Apress.,