ABSTRAK

Pada saat ini banyak algoritma kriptografi bermunculan, baik itu algoritma yang benar – benar baru, atau algoritma penyempurnaan algoritma yang sebelum nya dirasa sudah tidak cukup kuat untuk mengamankan isi dari *file* nya. Fitur seperti JCE (*Java Cryptography Extension*) masih tidak begitu familiar bagi para programmer sebagai *plugin* yang nanti nya akan sangat membantu dalam membuat sebuah *software* yang didalam nya menggunakan algoritma kriptografi. JCE sendiri hampir mendukung semua algoritma yang beredar saat ini, dari mulai algoritma klasik dan modern, simetris dan asimetris. Nantinya di tulisan ini akan dilakukan pengujian dengan cara membuat program dengan memanfaatkan JCE untuk empat algoritma, yaitu aloritma AES, DES 3DES, dan *Blowfish.* Untuk pengujian dilakukan dengan cara melakukan enkrispi dan dekripsi untuk file yang sama, maksud nya adalah tipe file dan ukuran file akan sama untuk semua algoritma. Setelah itu maka akan diambil data berupa lama waktu enkripsi dan dekripsi, dan banyak nya memori yang digunakan dalam proses enkripsi dan dekripsi. Nanti nya akan diketahui algoritma mana yang paling lambat atau pun yang paling cepat dalam hal proses enkripsi maupun dekripsi, dan juga dapat diketahui algoritma mana yang menggunakan memori paling banyak untuk proses enkripi maupun dekripsi.

Pada proses pengambilan data untuk masing – masing *file* didapatkan bahwa algoritma 3DES yang paling lambat dalam proses enkripsi maupun dekripsi. Ini berlaku untuk semua tipe *file* yang diproses, algoritma 3DES menempati urutan pertama dan selisih nya lumayan banyak dengan algoritma yang lain. Hampir tiga kali lipat dengan posisi ke dua, yaitu algoritma DES. Untuk algoritma tercepat di tempati oleh algoritma *Blowfish* dan terpaut sangat sedikit dengan algoritma AES. Untuk hasil *file* nya sendiri tidak mengalami *error*, artinya *file* awal identik dengan hasil *file* setelah mengalami proses enkripsi maupun dekripsi. *File* tidak mengalami redudansi data, ukuran nya sama dengan ukuran *file* awal. Untuk kinerja CPU sendiri akan mengalami proses peningkatan pada saat proses pembuatan *file* hasil enkripsi maupun dekripsi. Sedangkan pada saat proses enkripsi maupun dekripsi nya sendiri, kinerja CPU tidak menunjukan peningkatan proses yang berarti

ABSTRACT

At this time many cryptographic algorithms are springing up, be it the correct algorithm (the new right), or refinement algorithm before its not strong enough to secure the contents of the file . Features like JCE (*Java Cryptography Extension*) still not so familiar to programmers as a plugin even soon its be very helpful in making a software which in its use of cryptographic algorithms. JCE it self almost support all algorithms that are currently circulating, from getting classic and modern algorithms, symmetric and asymmetric. Later in this paper, the test will be done in a way to make the program by leveraging the JCE to four algorithm, which is AES, DES, 3DES and *Blowfish* algorithm. For testing performed by enkrispi and decryption for the same file, the intent is the file type and file size will be the same for all algorithms. After that it will be taken in the form of data encryption and decryption time, and much of his memory that will used in the encryption and decryption process. Later the result of the algorithms where the slowest or the fastest in terms of encryption or decryption, and can also be known algorithm which uses the most memory for the process of decryption or encryption.

On the data retrieval process for each file is obtained that the slowest is 3DES algorithm in encryption or decryption process. This applies to all types of files are processed, the algorithm is 3DES ranks first and the difference in his tolerable much with the other algorithms. Almost tripled with two positions wich is DES algorithm. The fastest algorithm is Blowfish and very little to be embedded with the AES algorithm. For the results of its own files is not experiencing an error, this means that the files are identical to the results of the initial file encryption process or after decryption. The files are the same size, data redundancies is equal to the size of the default file. For CPU performance alone will experience a process of improving upon the process of making *a* file encryption or decryption results. At the time of encryption or decryption, CPU performance does not indicate an increase in the process