APLIKASI PENGAMANAN DOKUMEN OFFICE DENGAN ALGORITMA KRIPTOGRAFI KUNCI ASIMETRIS ELGAMAL

Eko Aribowo
Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Ahmad Dahlan Yogyakarta
Email : ekoab@uad.ac.id, ekoab@yahoo.com

ABSTRAK

Makalah ini memaparkan mengenai pengembangan aplikasi kriptografi (enkripsi dan dekripsi) yang menerapkan algoritma asimetris Elgamal. Algoritma Elgamal dipilih karena algoritma ini merupakan salah satu algoritma asimetris yang cukup baru dan yang patennya baru dibuka 1997. Hasil pengembangan aplikasi ini diharapkan dan ditujukan untuk mendasari penelitian berikutnya tentang analisis algoritma kritografi asimetris. Selain itu hasil pengembangan ini juga ditujukan untuk pengayaan praktikum dalam mata kuliah Kriptografi di program studi Teknik Informatika. Untuk itu dalam mendukung 2 tujuan tersebut maka dibuat aplikasi kritpografi yang mana objek data digital yang digunakan adalah file-file dalam format office.Sehingga dengan aplikasi tersebut berbagai tujuan lain berikutnya dapat dicapai.

Kata kunci : Aplikasi, Elgamal, Kriptografi.

1. PENDAHULUAN

Kemajuan sistem informasi banyak sekali memberikan keuntungan dalam dunia bisnis, selain itu ada juga aspek-aspek dari sisi negatif dari kemajuan sistem informasi tersebut. Sebagai *end user computing* hampir semua aspek masyarakat menggunakan sistem informasi berbasis komputer, apalagi informasi-informasi mudah didapat dengan adanya jaringan komputer seperti LAN (*local Area Network*) dan internet memungkinkan menyediakan informasi secara cepat dan akurat [7].

Proses pengiriman data yang dilakukan media seperti *Local Area Network* (LAN), *internet, email, handphone* maupun media lain, pada dasarnya melakukan pengiriman data tanpa melakukan pengamanan terhadap konten dari data yang dikirim, sehingga ketika dilakukan penyadapan pada jalur pengirimannya maka data yang disadap dapat langsung dibaca oleh penyadap. Untuk menghidari kemungkinan data yang disadap dapat langsung dibaca oleh penyadap, maka data yang dikirim diacak dengan menggunakan metode penyandian tertentu sehingga pesan yang terkandung dalam data yang dikirim tersebut menjadi lebih aman [8].

Untuk menjaga keamanan dan terutama bagi suatu perusahaan, institusi atau organisasi yang mempunyai dokumen - dokumen rahasia dan penting. Mereka mengamankan dokumen - dokumen tersebut agar terhindar dari gangguan orang lain. Saat ini, sebagian besar dokumen - dokumen menggunakan aplikasi *Microsoft Word*. Sebagian besar terbiasa dengan aplikasi *Microsoft Office* yang sangat memudahkan siapa saja ketika menggunakan aplikasi ini. Pengolah kata *Microsoft Word*, begitu mudah digunakan sehingga siapapun yang menggunakannya akan merasa nyaman dengan pengolah kata ini. Dalam aplikasi *Microsoft Office* pengolah kata disimpan sebagai *file Microsoft Word*, pengolah angka sebagai *file Microsoft Excel*, dan sebagainya. Memang tidak ada yang aneh dalam sistem penyimpanan seperti ini karena memang sebagian besar di antara kita menggunakan semua aplikasi yang ada pada *Microsoft Office* [11].

Salah satu teknik untuk pengamanan data adalah dengan menggunakan algoritma penyandian data. Algoritma penyandian data saat ini semakin banyak jumlahnya, sejalan dengan berkembangnya ilmu yang mempelajari penyandian data tersebut. Ilmu ini biasa disebut Kriptografi [6].

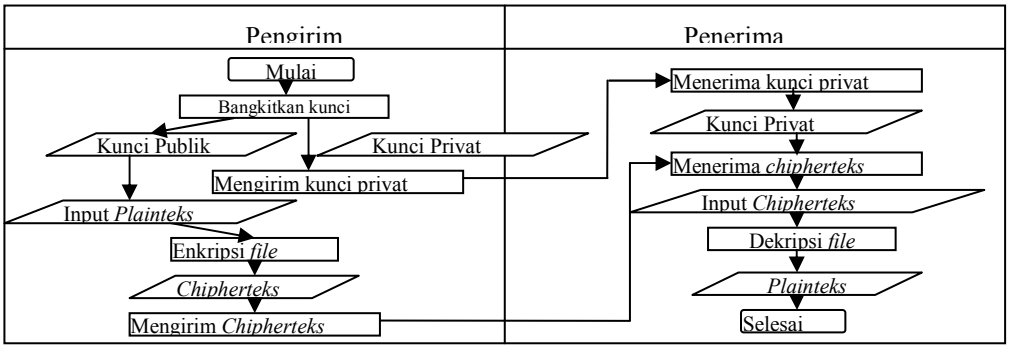
Dalam kriptografi terdapat beberapa metode yang cukup penting dalam pengamanan data, untuk menjaga kerahasiaan suatu data salah satunya adalah enkripsi (*encryption*). Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi *ciphertext*. Sedangkan suatu proses yang dilakukan untuk mengubah pesan tersembunyi menjadi pesan biasa (yang mudah dibaca) disebut dekripsi. Pesan biasa atau pesan asli disebut *plaintext* sedangkan pesan yang telah diubah atau disandikan supaya tidak mudah dibaca disebut dengan *cpihertext* [6].

Aplikasi keamanan data ditujukan untuk membantu mengatasi masalah keamanan data yang dibuat atau disimpan menggunakan aplikasi pada *Microsoft Office* sebagai contoh dokumen *Word, Excel* dan lain-lainnya dari pencurian dokumen-dokumen baik yang tidak penting maupun yang penting dan rahasia. Sehingga orang lain tidak dapat mengetahui isi dari dokumen-dokumen tersebut.

2. METODOLOGI PENELITIAN

2.1. Perancangan *Flowchart* dan Algoritma

Perancangan *flowchart* dan a*lgoritma* ditujukan untuk mempermudah pembuatan program. Dalam penelitian ini *flowchart* dan *algoritma* dibuat untuk mengetahui langkah-langkah apa saja yang harus diterapkan, dalam bahasa pemrograman, agar sistem yang dibuat dapat menghasilkan *output* yang sesuai dengan harapan dari *input*an yang dimasukkan oleh *user* berupa hasil dari proses enkripsi dan dekripsi *file* yang di*input*kan.



Gambar 1. *Flowchart* Sistem dengan *Algoritma* ElGamal

Skenario berjalannya *Flowchart* sistem pada gambar 1 adalah sebagai berikut :

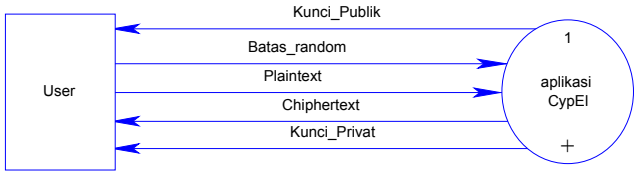
1. Mulai menjalankan sistem.
2. Langkah awal dari perancangan sistem adalah pengirim melakukan proses bangktikan kunci yang outputnya adalah kunci publik dan kunci privat.
3. Dari proses bangkitkan kunci dikirim kunci privat kepada penerima pesan.
4. Sedangkan kunci publik digunakan pada proses enkripsi oleh pengirim pesan. Masukkan kunci publik dan masukkan *plainteks*. Untuk kemudian di proses dalam proses enkripsi. Outputnya adalah *chiperteks*.
5. Langkah selanjutnya mengirim *chiperteks* kepada penerima pesan.
6. Untuk proses dekripsi, pertama kali penerima pesan input kunci privat dan masukkan *chiperteks*. Kemudian di proses dalam proses dekripsi. Outputnya adalah *plainteks*.
7. Sistem telah selesai dijalankan.

Sedangkan *algoritma* untuk *generate key*, enkripsi dan dekripsi masing-masing dijabarkan seperti *dilampiran*.

2.2. Perancangan Diagram Konteks

Diagram konteks merupakan penggambaran dari seluruh sistem yang akan dibuat. Tahap ini akan memperjelas hubungan antara *input* yang diberikan oleh *user* ke sistem dan *output* dari sistem kepada *user*.

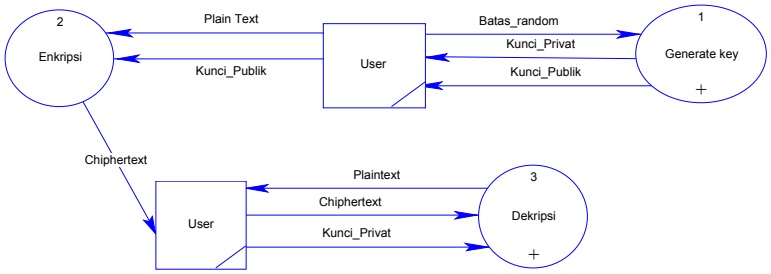
Dalam perancangan sistem ini akan terdapat sebuah entitas yaitu *user* yang terlebih dahulu memasukkan batasan random yang kemudian dilanjutkan dengan proses generate bilangan random untuk membangkitkan bilangan prima dan mendapatkan kunci publik *(public key)* dan kunci privat *(private key)* yang akan digunakan pada proses enkripsi dan dekripsi. Setelah itu *user* memberikan masukan ke sistem yang berupa *file* atau dokumen yang berekstensi .doc, .txt, .rtf, .xls, .mdb dan .ppt dalam bentuk *plaintext* ataupun *ciphertext*. Dari hasil masukan yang diberikan oleh *user*, sistem akan memberikan *output* berupa kunci publik *(public key)* dan kunci privat *(private key)* yang akan digunakan pada proses enkripsi dan dekripsi serta output berupa dokumen dari hasil enkripsi (*ciphertext*) atau dekripsi (*plaintext*).



Gambar 2. Diagram Konteks

2.3. Perancangan DFD (*Data Flow Diagram*)

Setelah merancang sebuah diagram konteks, kemudian akan dirancang DFD yang akan menjadi penggambaran dari seluruh sistem yang akan dibuat. DFD adalah sebuah cara untuk memodelkan alur data yang terjadi pada sebuah sistem. Dari diagram konteks yang telah dibuat diatas kemudian akan dijelaskan dengan penggambaran sebuah diagram alir data. Diagram ini akan menjelaskan bagaimana proses yang terjadi sebelum sebuah *file* atau dokumen yang dimasukkan oleh *user* menjadi *output* yang berupa *file* atau dokumen dari hasil enkripsi atau dekripsi.



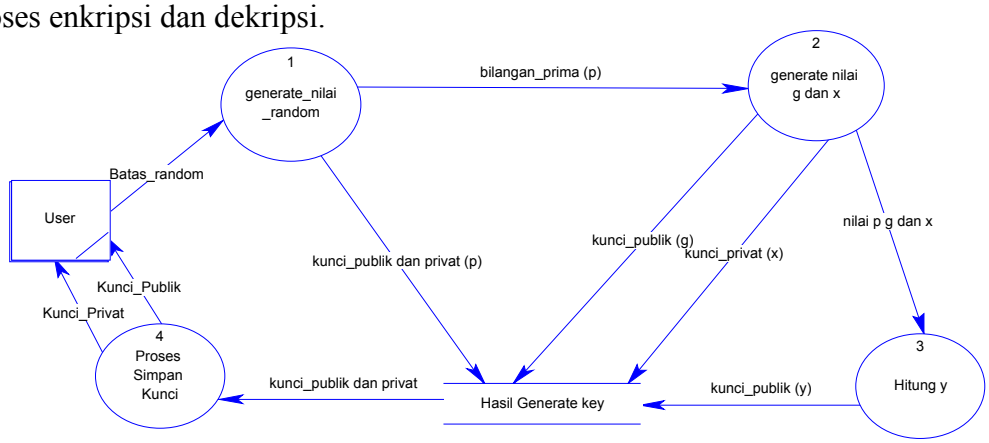
Gambar 3. DFD Level 0

Diagram alir data diatas menjelaskan alur data yang terjadi pada sistem, mulai dari *user* memasukkan batas random yang kemudian dilanjutkan dengan proses generate bilangan random untuk mendapatkan kunci publik *(public key)* dan kunci privat *(private key)* yang akan digunakan pada proses enkripsi dan dekripsi data, *user* sebagai pengirim pesan melakukan input yang berupa kunci publik dan *file* atau dokumen, menjadi *chipherteks* hasil enkripsi. Dan *user* sebagai penerima pesan menerima *chipherteks* hasil enkripsi dengan melakukan input yang berupa kunci publik dan *file* atau dokumen, menjadi *plainteks* hasil dekripsi. Alur data yang terjadi pada penggambaran DFD level 0 diatas adalah sebagai berikut:

a. *User* memasukkan batas random yang akan digenerate. Pada proses 1.1 hasil masukan batas random yang diberikan oleh *user* akan menghasilkan kunci publik *(public key)* dan kunci privat *(private key)* untuk *user* sebelum melakukan proses enkripsi dan dekripsi.
b. *User* sebagai pengirim pesan memasukkan kunci publik berupa nilai *y, g* dan *p* dan memasukkan *plaintext* atau dokumen, kemudian pada proses 1.2 hasil masukan *file* yang diberikan oleh *user* akan dienkripsi. Proses enkripsi akan menghasilkan *output file* enkrip berupa *ciphertext* kepada *user*.

Apabila *user* sebagai penerima pesan mendekripsi *file* maka data dari *user* berupa *password* untuk mengambil pasangan kunci privat berupa nilai *x* dan *p* akan masuk ke proses 1.2 bersama dengan masukkan lain yang berupa *ciphertext* yang akan didekripsi. Setelah *password* dan *file* yang akan didekripsi dimasukkan maka dihasilkan *output file* dekrip berupa *plaintext* kepada *user*.

Dari Alur data yang terjadi pada penggambaran DFD level 0 diatas kemudian akan dijelaskan dengan penggambaran sebuah diagram alur data DFD level 1 untuk proses generate key. Diagram ini akan menjelaskan bagaimana proses yang terjadi sebelum mendapatkan kunci publik dan kunci privat yang akan digunakan pada proses enkripsi dan dekripsi.



Gambar 4. DFD Level 1 Proses *Generate Key*

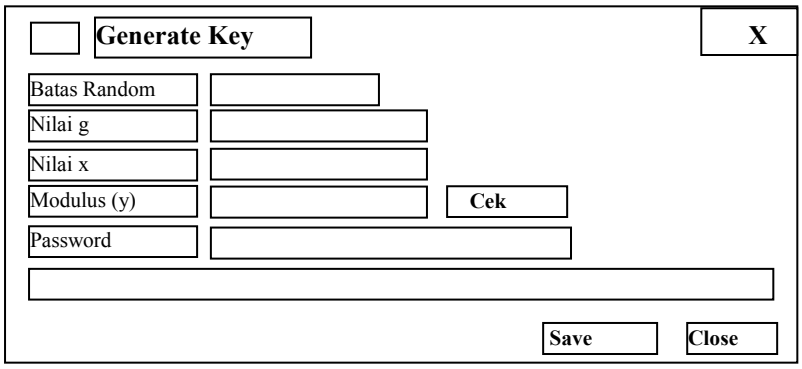
Alur data yang terjadi pada penggambaran DFD level 1 diatas adalah sebagai berikut:

User memasukkan batas random yang akan digenerate. Kemudian pada proses 1.1.1 nilai random inputan digunakan untuk membangkitkan satu bilangan prima (*p*). Bilangan prima (*p*) digunakan untuk membangkitkan nilai *g* dan *x* pada proses 1.1.2 serta menghitung nilai *y* pada proses 1.1.3. Kemudian, nilai p, *g, x* dan *y* disimpan dalam tabel *key*. Dimana nilai *y, g, p* adalah kunci publik *(public key)* dan pasangan kunci *x, p* sebagai kunci privat *(private key)*. Pada proses 1.1.4 masing-masing kunci disimpan, untuk keamanan maka *user* harus menyimpan kunci publik *(public key)* dan kunci privat *(private key)* dengan menggunakan *password*. Pada proses ini menghasilkan kunci publik *(public key)* dan kunci privat *(private key)* untuk *user* sebelum melakukan proses enkripsi dan dekripsi.

2.4. Perancangan Antar Muka (*Interface*)

Perancangan antar muka program aplikasi keamanan dokumen pada *Microsoft Office* proses enkripsi dibuat dengan menggunakan *Borland Delphi* 7.0 karena pada *Delphi* telah tersedia *tool-tool* yang dipergunakan dalam implementasi

program. Sehingga akan memudahkan dalam mendesain antar muka maupun dalam implementasi sistem.



Gambar 5. Rancangan *Form Generate Key*

3. HASIL DAN PEMBAHASAN

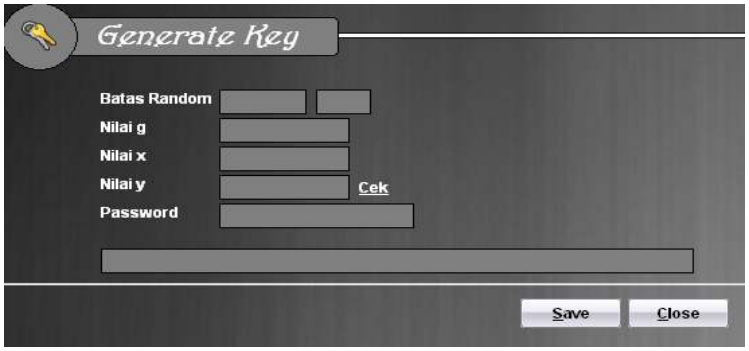
Dalam pembuatan aplikasi ini diperlukan enam *form* seperti yang telah dirancang diatas, yaitu *form* utama, *form generate key*, *form* proses, *form* laporan, *form* help *dan form* about system.

3.1. *Form* Utama

Form utama menampilkan menu utama dari aplikasi. *Form* ini merupakan tampilan awal aplikasi berfungsi sebagai tempat untuk menampilkan semua menu-menu yang dibuat dalam aplikasi ini. Dalam *form* ini ditampilkan informasi tentang aplikasi dengan judul Kriptografi ElGamal untuk Pengamanan Dokumen Office. Disediakan beberapa menu pilihan yaitu menu "*Generate key"*, "Proses", "Laporan", "Help", "about System", "date and time" dan "Exit". *User* dapat dengan mudah melakukan pilihan menu yang akan dibukanya.

3.2. *Form Generate Key*

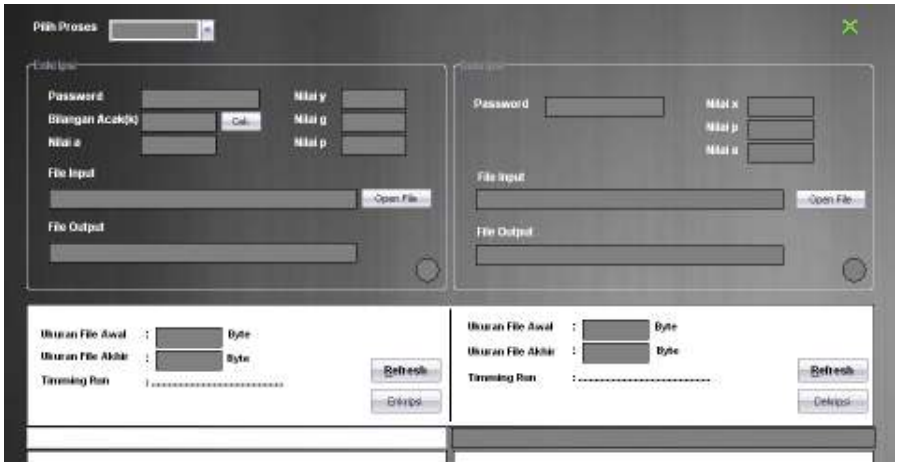
Form ini digunakan untuk mendapatkan nilai *key* (kunci) yang digunakan pada proses enkripsi dan dekripsi. Memulai proses *generate key* masukkan batas random yang kemudian akan digenerate menghasilkan bilangan prima (*p*). Program akan mengecek bilangan yang dimasukkan prima atau tidak. Dari *generate* bilangan prima yang dihasilkan diperoleh juga nilai *g* dan nilai *x* karena syarat dari nilai *g* dan *x* adalah $g < p$ dan $1 \leq x \leq p - 2$. Masing-masing nilai dari *p,g,*dan x digunakan untuk menghitung nilai $y = g^X \bmod p$. Memperoleh nilai *y* dengan mengklik *button* cek. Setelah diperoleh nilai *y* maka *user* wajib untuk memasukkan *password*, demi kemanan kunci yang digunakan. Setelah semua data terisi, klik *button* save maka data masing-masing nilai disimpan dalam tabel *key*. Nilai *p, g, x* dan *y* sudah diperoleh maka dapat dikelompokkan masing-masing nilai yang termasuk dalam kunci publik (nilai *y, g, p*) dan yang termasuk dalam kunci privat (nilai *x, p*). Jika pengisian data belum lengkap maka ada peringatan dari program.



Gambar 6 . Tampilan *Form Generate Key*

3.3. *Form* Proses

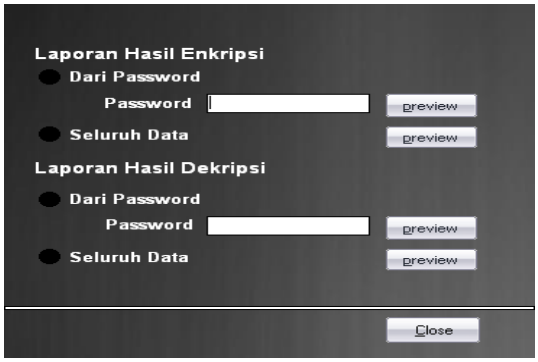
Form ini digunakan untuk menampilkan proses enkripsi dan dekripsi.User harus terlebih dahulu memilih proses yang akan dijalankannya. Pilih proses enkripsi atau proses dekripsi. Memulai proses enkripsi, ambil *public key* yang terdiri dari nilai *y, g, dan p* yang sudah tersimpan pada saat proses *generate key.* Bangkitkan sebuah bilangan acak *k* yang dalam hal ini $1 \leq k \leq p - 2$. Nilai *k* digunakan untuk menghitung nilai *a* dan *b* dengan rumus $a = g^k \bmod p$ dan $b = y^k m \bmod p$, pasangan *a* dan *b* adalah chiperteks (pesan rahasia). *Open File* yang akan dienkripsi dengan cara mengkilk *button open file*, secara otomatis akan tampil ukuran *file* awal. Klik *button* enkripsi ada permintaan untuk *save file* atau dokumen hasil enkripsi dengan tipe *file* atau dokumen yang sama, proses enkripsi berjalan. Setelah itu, tampil secara otomatis ukuran *file* akhir dan *timming run* proses enkripsinya. Memulai proses dekripsi, ambil *privat key* yang terdiri dari nilai *x* dan *p* yang sudah tersimpan pada saat proses *generate key. Open File* yang akan didekripsi dengan cara mengkilk *button open file*, secara otomatis akan tampil ukuran *file* awal. Klik *button* proses ada permintaan untuk *save file* atau dokumen hasil enkripsi dengan tipe *file* atau dokumen yang sama, proses dekripsi berjalan. Setelah itu, tampil secara otomatis ukuran *file* akhir dan *timming run* proses dekripsinya.



Gambar 7. Tampilan *Form* Proses

3.4. *Form* Laporan

F*orm* laporan digunakan untuk menampilkan hasil proses enkripsi dan dekripsi. Form ini terdiri dari 4 pilihan yaitu laporan data enkripsi dari password dan seluruh data serta laporan data dekripsi dari password dan seluruh data.



Gambar 8. Tampilan *Form* Laporan

3.5. Coding

Source code / program dari implementasi ini dapat dibaca secara lebih detail dalam lampiran. Coding yang ada antara lain untuk proses Generate key, cek bilangan prima, proses enkripsi dan dekripsi, konversi file office ke dalam nilai-nilai integer.

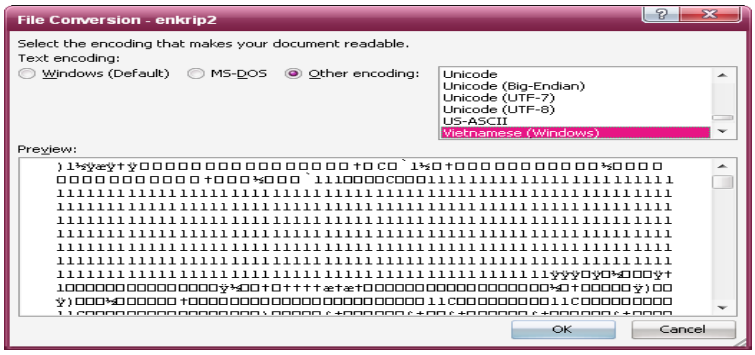
3.6. Pengujian

a. Pengujian Dokumen *Word*

Pengujian pada dokumen *Word* berekstensi .doc dengan ukuran 24576 byte adalah sebagai berikut :

Gambar 9. Dokumen *Plainteks* Berekstensi .DOC

Setelah dokumen plainteks diatas dienkripsi menggunakan aplikasi keamanan dokumen, maka dapat dihasilkan sebuah dokumen cipherteks dengan ukuran 54784 byte. Hasil dari enkripsi tersebut adalah sebagai berikut :

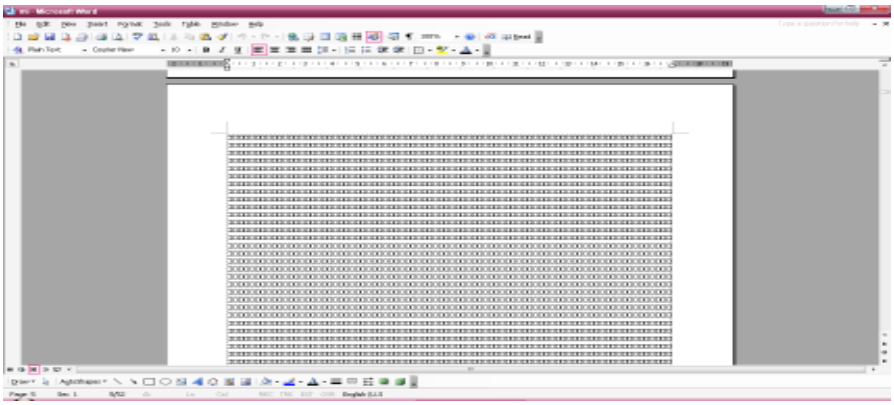


Gambar 10. Dokumen *Cipherteks* Bekstensi .DOC

Setelah pengujian pada dokumen berekstensi .doc, selanjutnya dilakukan pengujian pada dokumen .rft. Pengujian dilakukan pada sebuah dokumen dengan ukuran 18623 byte. Dokumen plainteks tersebut adalah sebagai berikut :

Gambar 11. Dokumen *Plainteks* Berekstensi .RTF

Setelah dokumen plainteks diatas dienkripsi menggunakan aplikasi keamanan dokumen, maka dapat dihasilkan sebuah dokumen cipherteks dengan ukuran 38670 byte. Hasil dari enkripsi tersebut adalah sebagai berikut :



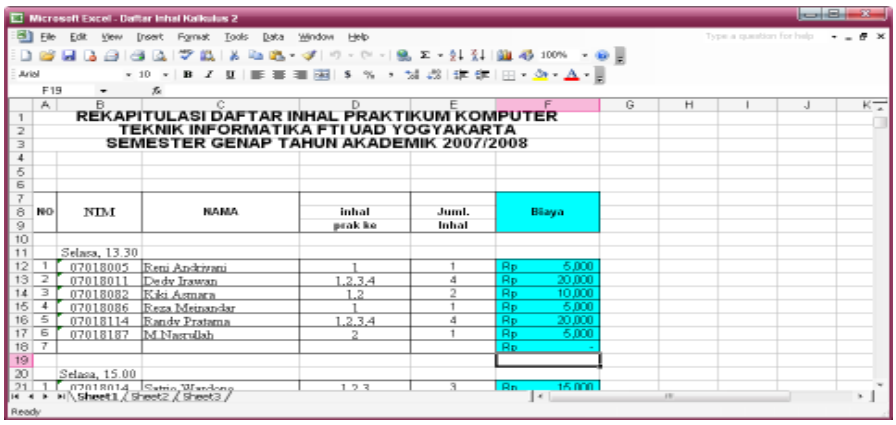
Gambar 12. Dokumen *Cierteks* berekstensi .RTF

Setelah pengujian pada dokumen berekstensi .doc dan .rtf, selanjutnya dilakukan pengujian pada dokumen .txt. Pengujian dilakukan pada sebuah dokumen dengan ukuran 1338 byte.

Setelah dokumen plainteks diatas dienkripsi menggunakan aplikasi keamanan dokumen, maka dapat dihasilkan sebuah dokumen cipherteks dengan ukuran 1338 byte. Hasil dari enkripsi tersebut adalah sduah dipastikan dalam bentuk *ciphertext*.

b. Pengujian Dokumen *Excell*

Setelah pengujian pada dokumen *Word* berhasil, selanjutnya dilakukan pengujian pada dokumen *Excell*. Pengujian dilakukan pada sebuah dokumen dengan ukuran 67072 byte dengan ekstensi .xls. Dokumen plainteks tersebut adalah sebagai berikut :

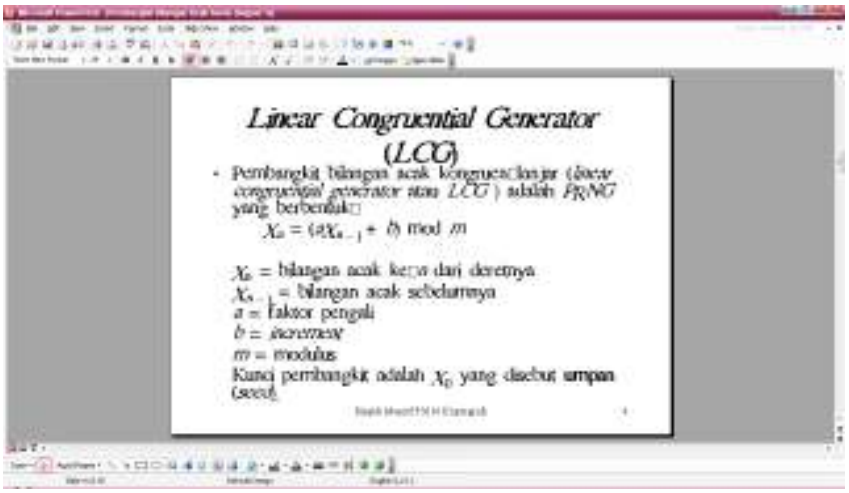


Gambar 13. Dokumen *Plainteks* Berekstensi .XLS

Dari pengujian tersebut didapat hasil sebuah dokumen cipherteks dengan ukuran 127119 byte, yagn tetap dalam format .xls, namun tidak dapat dibaca atau dengan isi *ciphertex.t*

c. Pengujian Dokumen *PowerPoint*

Pengujian pada dokumen *PowerPoint* dilakukan menggunakan dokumen berekstensi .ppt sebagai plainteksnya. Dokumen yang digunakan adalah dokumen dengan ukuran 105792 byte. Dokumen plainteks tersebut adalah sebagai berikut :



Gambar 14. Dokumen *Plainteks* Berekstensi .PPT

Dari pengujian tersebut didapat hasil sebuah dokumen cipherteks dengan ukuran 225839 byte, dengan bentuk dokumen PPT yang sudah ter-coding-kan, sehingga tidak dapat dibuka dengan aplikasi MS Powerpoint.

d. Pengujian Dokumen *Access*

Pengujian pada dokumen *Access* dilakukan menggunakan dokumen berekstensi .mdb sebagai plainteksnya. Dokumen yang digunakan adalah dokumen dengan ukuran 165888 byte. Dokumen plainteks tersebut adalah dalam format .MDB

Dari pengujian tersebut didapat hasil sebuah dokumen cipherteks dengan ukuran 265935 byte. dengan bentuk dokumen MDB yang sudah ter-coding-kan, sehingga tidak dapat dibuka dengan aplikasi MS Acces.

Semua dokumen yang telah dienkripsi akan kembali ke bentuk semula setelah dokumen-dokumen tersebut didekripsi dan proses dekripsi tersebut tidak mengubah format dari dokumen-dokumen tersebut.

4. SIMPULAN

Dari langkah-langkah yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut :

1. Proses pengujian aplikasi keamanan dokumen dengan metode *black box test* dan *alpha test*. Aplikasi diujikan kepada responden, hasilnya adalah aplikasi ini sangat bermanfaat untuk mengamankan dokumen agar isinya tidak diketahui oleh orang lain yang tidak berhak.
2. Implementasi program ini menghasilkan suatu aplikasi yang dapat mengubah isi suatu dokumen (plainteks) yang berupa teks, tabel dan gambar menjadi kode-kode yang tidak dikenal (cipherteks). Mengembalikan isi dari dokumen dari kode-kode (cipherteks) menjadi dokumen aslinya (plainteks).
3. Aplikasi yang dihasilkan dapat digunakan untuk dokumen office dengan ekstensi .doc, .txt, .rtf, .xls, .ppt dan .mdb.

DAFTAR PUSTAKA

[1] Andari, Yuli, 2007, *Implementasi Pengamanan Dokumen Pada Microsoft Office Dengan Algoritma Kriptografi RC4 Stream Chiper dan SHA-1*, Skripsi S-1, Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta.

[2] Aribowo,Eko, 2002, *Organisasi Berkas*,Teknik Informatika, Universitas Ahmad Dahlan Yogyakarta.

[3] Husni, 2004, *Pemrograman Database dengan Delphi,* Graha Ilmu, Yogyakarta.

[4] Khudri, Wan, 2005 , *Enkripsi dan Deskripsi Data Menggunakan Algoritma ElGamal ECC,* Skripsi S-1, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret, Surakarta.

[5] Madcoms, 2002, *Pemrograman Borland Delphi 7*, Andi Offset, Yogyakarta.

[6] Munir, Rinaldi, 2006, *Kriptografi*, Informatika, Bandung.

[7] Pressman, RS, 1992, *Software Engineering A Practitioner's Approach*, The Mc Graw-Hill Companies, New york.

[8] Raharjo, Budi, 1999, " *Keamanan Sistem Informasi Berbasis Internet* ", PT. Insan Komunikasi, Indonesia.

[9] Syaifudin, Nur, 2006, *Implementasi Algoritma Kriptografi Blowfish Untuk Keamanan Dokumen Pada Microsoft Office*, Skripsi S-1, Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta.
[10] Wahana, Komputer, 2003, *Panduan Praktis Pemrograman Borland Delphi 7*, Andi Offset, Yogyakarta.
[11] Yusuf, Kurniawan, "*Kriptografi Keamanan Internet dan Jaringan Komunikasi*", Informatika Bandung, 2004.
[12] _____, 11 Agustus 2003, " *Aplikasi yang Bertenaga dan Komprehensif* ", http://kompas.com/kompas-cetak/0308/11/tekno/485174.htm/ 26 April 2008
[13] Http:// www.cert.or.id/~budi/courses/ec7010/2004-2005/nana-report.doc/ 28 Februari 2008
[14] Http:// ilmu-komputer.net/algorithms/sejarah-kriptografi/ 30 April 2008
[15] Http:// zakimath.jogjainfo.com/2007/07/04/algoritma-kriptografi/ 03 Mei 2008
[16] Http:// mail.uns.ac.id/~chudry/pdf/I.pdf/ 27 Maret 2008
[17] Http://zakimath.jogjainfo.com/2007/07/04/algoritma-elgamal/ 30 Maret 2008