ISSN : 2598–3814 (Online), ISSN : 1410–4520(Cetak)

PERANCANGAN APLIKASI PENGAMAN DOKUMEN DENGAN ALGORITMA XXTEA

Haikal Nando Winata, Bahtera Kelana, Selly Annisa, Beni satria, Mery Sri Wahyuni, Zulkarnain Lubis, Abdullah Muhazzir
Jurusan Teknik Informatika Institut Teknologi Medan
Email : dr.zulkarnainlubis@itm.ac.id

Abstrak

Pencurian data yang terkandung dalam dokumen digital dapat dihindari dengan cara mengamankan file dokumen dengan menggunakan kriptografi. XXTEA sebagai salah satu algoritma kriptografi yang berbasis pada iterasi feistel merupakan solusi untuk masalah ini. Tanpa perhitungan yang rumit, algoritma XXTEA menghasilkan file dokumen teracak yang tidak dapat diakses tanpa harus melakukan proses dekripsi terlebih dahulu terhadap file tersebut. Penelitian ini menghasilkan aplikasi pengaman file dokumen yang memanfaatkan algoritma XXTEA dan bahasa pemrograman Microsoft Visual Basic .Net 2010.

Kata-Kata Kunci: *Kriptografi, File Dokumen, XXTEA, Microsoft Visual Basic .Net 2010*

I. Pendahuluan

Pesatnya perkembangan teknologi komputer saat ini sering mengakibatkan penyalahgunaan teknologi tersebut dalam tindakan kriminal. Salah satu yang paling sering terjadi adalah pencurian data yang terkandung dalam dokumen digital. Untuk itu, perlu dilakukan suatu bentuk pengamanan terhadap hal tersebut guna meningkatkan rasa aman bagi pemilik dokumen digital yang bersangkutan.

II. Tinjauan Pustaka

2.1 Dokumen Digital

Dokumen merupakan suatu sarana transformasi informasi dari satu orang ke orang lain atau dari suatu kelompok ke kelompok lain. Dokumen meliputi berbagai kegiatan yang diawali dengan bagaimana suatu dokumen dibuat, dikendalikan, diproduksi, disimpan,didistribusikan, dan digandakan. Dokumen sangat penting, baik dalam kehidupan sehari-hari, organisasi, maupun bisnis.

2.1.2 Format Dokumen Digital

1. RTF (Rich Text Format)
2. DOC (*Document*)
3. PDF (*Portable Document Format*)

2.2 Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.

2.3 Kriptografi

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematis yang berhubungan dengan aspek keamanan informasi.

2.3.2 Algoritma Kriptografi

1. Algoritma kriptografi simetris
2. Algoritma kriptografi asimetris

2.4 XXTEA

Metode *corrected block tiny encryption* atau lebih dikenal dengan nama XXTEA adalah sebuah algoritma penyandian yang sederhana, tapi kuat yang berbasis iterasi Feistel dan menggunakan banyak ronde untuk mendapatkan keamanan.

2.5 Alat Bantu Perancangan Sistem

Adapun alat bantu yang sering digunakan dalam perancangan atau pengembangan program yang digunakan dalam penelitian adalah *Flowchart* dan *Activity Diagram.*

2.6 *Microsoft Visual Studio 2010*

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET *Framework*, dengan menggunakan bahasa BASIC.

2.6.1 Sejarah Visual Basic 2010

Pada tahun 1997 Visual Basic 5.0 dirilis dengan memasukkan teknologi baru yang mendukung COM serta memungkinkan membuat kontrol ActiveX sendiri ataupun DLL. Pada tahun 1998 Visual BAsic 6.0 dirilis dengan teknologi yang lebih ditingkatkan lagi khususnya dalam mengakses SQL Server dan mengusung ADO (ActiveX Data Object) yang dirancang untuk meningkatkan kinerja dalam mengakses database pada perusahan besar.

2.6.2 Lingkungan Kerja pada Microsoft Visual Basic 2010

1. Toolbox
2. Solution Explorer
3. Properties Window

III. Metode Penelitian

3.1 Metode Pengumpulan Data

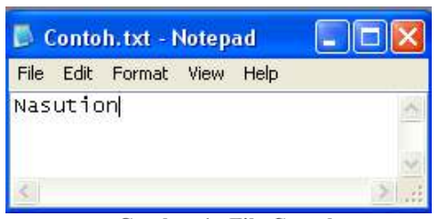
Adapun metode pengumpulan data yang penulis lakukan adalah berupa teori mengenai kriptografi dan cara kerja algoritma XXTEA yang dikumpulkan dari sumber buku bacaan serta jurnal penelitian.

3.2 Analisa Sistem

Dalam perancangan aplikasi pengaman dokumen menggunakan algoritma XXTEA ini, dilakukan analisa terhadap cara kerja algoritma XXTEA dalam melakukan enkripsi dan dekripsi.

- Proses Enkripsi

Dalam analisa proses enkripsi algoritma XXTEA ini, diambil contoh kasus proses enkripsi sebuah file TXT dengan nama Contoh.txt yang berisi "Nasution" seperti terlihat pada Gambar 1, dengan kata kunci "Enkripsi".



Gambar 1. File Contoh.txt

Langkah pertama yang dilakukan adalah mengambil nilai *byte* dari *file* tersebut sehingga diperoleh deret *byte* sebagai berikut:

78 97 115 117 116 105 111 10
69 110 107 114 105 112 115 105

Langkah berikutnya adalah mengambil nilai *binary* dari *byte* pada posisi (r+n−1) mod n dan posisi (r+1) mod n, sehingga diperoleh hasil sebagai berikut:

X_{r-1} = Binary(X((1+7) mod 16)
= Binary (X(8))
= 00001010
X_{r+1} = Binary(X((1+1) mod 16)
= Binary (X(2))
= 01100001

Lakukan pengacakan dengan menghitung X_{r-1}<<2 XOR X_{r+1}>>5,

Y_1 = X_{r-1}<<2 XOR X_{r+1}>>5
= 00101000 XOR 00001011
= 00100011

Lakukan pengacakan dengan menghitung X_{r-1}>>3 XOR X_{r+1}<<4,

Y_2 = X_{r-1}>>3 XOR X_{r+1}<<4
= 01000001 XOR 00010110
= 01010111

Jumlahkan Y_1 dan Y_2.

Z_1 = $Y_1 + Y_2$
= 01000001 XOR 00010110
= 01010111

Lakukan pengacakan dengan menghitung X_{r-1} XOR D, dengan D adalah nilai binary dari Delta (0x9E3779B), sehingga diperoleh hasil sebagai berikut:

D = Binary(0x9E3779B)
= Binary(2654435769)
=
10011110001101110111100110111001
Y_3 = X_{r-1} XOR D
= 00101000 XOR
10011110001101110111100110111001
=
10011110001101110111100110010001

Lakukan pengacakan dengan menghitung X_{r-1} XOR Binary dari K(E), dengan E adalah nilai desimal dari D>>2 AND 3, sehingga diperoleh hasil sebagai berikut:

E = Desimal(D>> AND 3)
= Desimal
(01100111100011011101111001101110 AND 3)
= Desimal(00000010)
= 2
Y_4 = X_{r+1} XOR Binary(K(2))
= 01100001 XOR Binary(110)
= 01100001 XOR 01101110
= 00001111

Jumlahkan Y_3 dan Y_4 sehingga diperoleh hasil sebagai berikut:

Z_2 = $Y_3 + Y_4$
=
10011110001101110111100110010001 + 00001111
=
10011110001101110111100110100000

Hitung nilai desimal dari Z1 XOR Z2, dengan hasil sebagai berikut:

C_1 = Desimal(Z_1 XOR Z_2)
= Desimal(01010111 XOR 10011110001101110111100110100 000)
=
Desimal
(10011110001101110111100111110111)
= 2654435831

Proses enkripsi dilanjutkan untuk nilai *byte* pada posisi selanjutnya hingga seluruh nilai *byte* dienkripsi.

-Proses Dekripsi

Untuk proses dekripsi metode XXTEA, merupakan proses yang sama dengan proses enkripsi. Namun, berbeda dengan proses enkripsi dimana proses dimulai dari posisi *byte* pertama hingga jumlah deret *byte*, proses dekripsi dimulai dari *byte* terakhir dari deret *byte* hingga *byte* pada posisi pertama.

3.3 Rancangan Penelitian

Dalam perancangan aplikasi pengaman dokumen digital dengan algoritma XXTEA ini, tahapan perancangan yang dilakukan terbagi menjadi dua bagian, yaitu rancangan proses serta rancangan antarmuka program.

3.3.1 Rancangan Proses

Berdasarkan hasil analisa sistem yang dilakukan, dirancang sebuah alur proses yang akan digunakan dalam perancangan aplikasi. *Flowchart* yang dirancang untuk proses enkripsi dan dekripsi dokumen digital menggunakan algoritma XXTEA ini di mulai dari munculnya form enkripsi dan dekripsi ketika sistem pertama kali dijalankan, kemudian apabila user memilih mode enkript maka sistem akan melakukan enkripsi teks yang terdapat dalam surat elektronik sesuai dengan metode yang dipilih yaitu metode XXTEA, sedangkan apabila memilih menu dekript maka akan dilakukan proses sebaliknya yaitu proses dekripsi teks yang telah di enkripsi sebelumnya.

3.3.2 Rancangan Antarmuka

Setelah merancang proses kerja perangkat lunak yang akan dirancang, selanjutnya dilakukan perancangan terhadap antarmuka perangkat lunak ini. Dalam perancangan ini, penulis merancang empat form yang dapat digunakan pengguna untuk berinteraksi dengan perangkat lunak yang dirancang.

IV. Hasil Dan Pembahasan

4.1 Kebutuhan Spesifikasi Minimum

Agar dapat berjalan dengan baik, ada beberapa spesifikasi minimum yang harus dipenuhi pada sistem yang dirancang ini. Adapun spesifikasi tersebut dibagi menjadi dua bagian, yaitu spesifikasi *hardware* dan spesifikasi *software*.

4.2 Pengujian Aplikasi

Dari hasil implementasi perancangan program, diperoleh sebuah aplikasi pengaman dokumen dengan menggunakan metode XXTEA. Aplikasi ini kemudian diuji untuk melihat apakah sudah dapat berjalan dengan baik dan sesuai dengan bentuk rancangan sebelumnya.

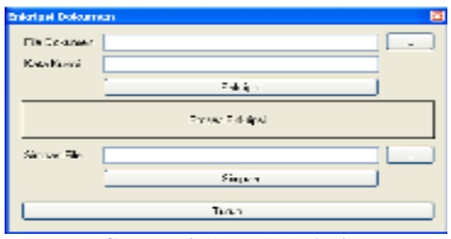
-Pengujian form Utama

Pengujian ini dilakukan untuk melihat apakah form Utama yang dihasilkan sudah dapat bekerja dengan baik sesuai dengan bentuk rancangan sebelumnya. Pengujian ini dilakukan dengan melakukan eksekusi terhadap aplikasi sehingga muncul tampilan form Utama seperti terlihat pada Gambar 2.

Gambar 2. Form utama

-Tampilan Form Utama

Selanjutnya dilakukan penekanan tombol Enkripsi untuk melihat apakah form Enkripsi sudah dapat ditampilkan melalui form Utama. Hasil yang diperoleh adalah munculnya tampilan form Enkripsi, seperti terlihat pada Gambar3.



Gambar 3. Form Enkripsi

-Tampilan Form Enkripsi

Selanjutnya, dilakukan penekanan tombol Dekripsi pada form Utama untuk melihat apakah form Dekripsi sudah dapat diakses melalui form Utama. Hasil yang diperoleh adalah munculnya form Dekripsi seperti terlihat pada Gambar 4.

Gambar 4. Form dekripsi

-Tampilan Form Dekripsi

Selanjutnya dilakukan penekanan tombol ”?” pada form Utama untuk melihat apakah form Info sudah dapat diakses dari form Utama. Hasil yang diperoleh adalah munculnya form Info, seperti terlihat pada Gambar 5.

Gambar 5. Form info

1. Pengujian Enkripsi

Pengujian ini dilakukan untuk melihat apakah aplikasi telah dapat melakukan enkripsi terhadap dokumen sesuai dengan bentuk rancangan sebelumnya. Dalam pengujian ini, digunakan dokumen dengan format TXT dengan nama file

Uji1.txt. Pengujian dimulai dengan mengakses form Enkripsi, menginputkan file Uji1.txt dan kata kunci ”Enkripsi”, sebagaimana terlihat pada Gambar 6.

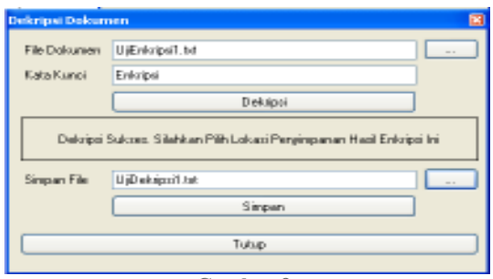
Gambar 6. Form Enkripsi

Input File Uji1.txt dan Kata Kunci

Selanjutnya, dilakukan penekanan tombol Enkripsi, sehingga muncul pesan bahwa proses enkripsi sukses, seperti terlihat pada Gambar 7.

Gambar 7.

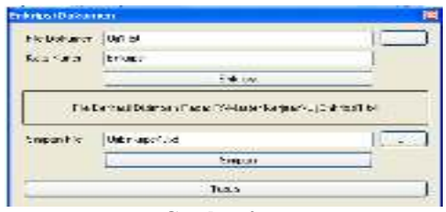
Dekripsi Sukses selanjutnya, dilakukan pemilihan lokasi penyimpanan file hasil dekripsi. Pada pengujian ini, hasil dekripsi di atas disimpan dengan nama file UjiDekripsi1.txt.



Gambar 8.

Penyimpanan Hasil Dekripsi

Selanjutnya dilakukan penekanan tombol Simpan untuk menyimpan file hasil dekripsi tersebut pada lokasi yang sudah dipilih sebelumnya, sehingga muncul tampilan pesan bahwa proses penyimpanan sukses, sebagaimana terlihat pada Gambar 9.



Gambar 9.

4.3 Pembahasan

Dari hasil pengujian aplikasi yang dilakukan, diperoleh beberapa kelebihan dan kelemahan dari sistem yang dihasilkan. Adapun kelebihan dan kelemahan dari sistem yang diperoleh adalah sebagai berikut:

4.3.1 Kelebihan Sistem

- -Sistem dapat menyimpan file dokuemen dengan format yang sama seperti file awal yang diinputkan, baik untuk proses enkripsi maupun dekripsi.
- -Proses enkripsi yang mengacak struktur file dokumen menyebabkan file tersebut tidak dapat diakses oleh aplikasi pengolah kata yang bersangkutan.

4.3.2 Kelemahan Sistem

Sistem masih memiliki ketebatasan dalam hal format *file* dokumen yang dapat diproses. *File* yang dapat diproses oleh aplikasi ini terbatas pada file dengan format TXT, DOC, DOCX dan PDF..Sistem belum dapat melakukan enkripsi terhadap lebih dari satu file dokumen secara bersamaan.

V. Kesimpulan Dan Saran

5.1 Kesimpulan

1. Dengan mengimplementasikan algoritma XXTEA, dihasilkan sebuah aplikasi yang dapat mengamankan file dokumen, sehingga tidak dapat diakses dengan mudah oleh pihak lain yang tidak memiliki hak untuk mengaksesnya.
2. Hasil enkripsi dan dekripsi yang menyerupai format awal sebelum diproses merupakan kelebihan dari sistem yang dihasilkan dari implementasi algoritma XXTEA ini. Walaupun demikian, keterbatasan dalam hal format file dokumen yang dapat diproses merupakan kelemahan yang diharapkan dapat dikembangkan pada penelitian berikutnya.

5.2 Saran

1. Sistem ini dapat dikembangkan lebih lanjut dengan cara menambahkan format file dokumen yang dapat diproses, sehingga menambah daya guna dari aplikasi yang dihasilkan.
2. Untuk penelitian selanjutnya, dapat ditambahkan algoritma kriptografi yang lain sehingga dapat dibandingkan hasil yang diperoleh antara algoritma tersebut dengan algoritma XXTEA yang digunakan dalam penelitian ini.

Daftar Pustaka

[1] Ariyus, D., 2008, *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*, Penerbit Andi, Yogyakarta

[2] Hariyanto, B., 2009, *Sistem Operasi*, Informatika, Bandung.

[3] Jogiyanto, H.M., 2005, *Analisis & Disain Sistem Informasi Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*, Edisi ke-2, Andi Offset, Yogyakarta.

[4] Sianipar, R.H., 2014, *Pemrograman Visual Basic.NET*, Informatika, Bandung.

[5] Stalling, W., 2003, *Crypthography and Network Security*, Prentice-hall inc, Amsterdam

[6] Tanenbaum, A., 1990, *Structured Computer Organization*, Third edition, Prentice-hall inc, Amsterdam.

[7] Wheeler, D. J. dan Needham, R. M., 1998, *Correction To XTEA*, Cambridge University, England

[8] Zarlis, M. & Handrizal, 2008, *Algoritma dan Pemrograman : Teori dan Praktik Dalam Pascal*, Edisi Kedua,USU Press, Medan