APLIKASI KRIPTOGRAFI DENGAN METODE VERNAM CIPHER DAN METODE PERMUTASI BINER

M. Sholeh dan J.V. Hamokwarong

Teknik Infomartika
Fakultas Teknologi Industri
Institut Sains & Teknologi AKPRIND Yogyakarta
Jl. Kalisahak 28 Komplek Balapan Yogyakarta
Email :muhash@akprind.ac.id

Salah satu proses pengamanan yang dapat diterapkan dalam proses penyimpanan atau pengiriman file adalah dengan melakukan proses kriptografi. Proses kriptografi dilakukan dengan melakukan proses pengacakan data, sehingga file ang asli tidak mudah untuk dibaca oleh pihak ang tidak berkepentingan.

Banyak algoritma atau metode yang dapat digunakan untuk proses kriptografi.. Metode Vernam Cipher merupakan algoritma berjenis symmetric key kunci yang digunakan untuk melakukan enkripsi dan dekripsi yang menggunakan kunci yang sama. Dalam proses enkripsi, algoritma Vernam Cipher menggunakan cara stream cipher dimana cipher berasal dari hasil XOR antara bit plaintext dan bit key, sedangkan permutasi biner dilakukan dengan membalikan kode biner pada setiap karakter

Dalam makalah ini akan dibahas, program aplikasi yang dapat melakukan proses kriptografi terhadap suatu file. Proses kriptografi yang terdiri dari enkripsi dan dekripsi akan menggunakan metode Vernam Cipher dan metode permutasi biner. Program dikembangkan dengan menggunakan microsoft visual Basic.

***Kata Kunci** : Kriptografi, vernam cipher, permutasi biner*

Latar Belakang Masalah

Seiring perkembangan teknologi saat ini yang semakin pesat maka proses pengiriman data dapat dilakukan dengan mudah dan melalui berbagai macam media yang ada. Antara lain, memalui media *internet* dengan melakukan fasilitas *email*, melalui transfer data antar perangkat *mobile* (Handphone, PDA, Flash Disk) maupun dengan teknologi radio *frequency* hingga dengan menggunakan jaringan komputer.

Proses perkembangan yang pesat dalam proses pengiriman data membawa dampak yang sangat besar, yaitu masalah kemanan data yang di kirim. untuk itu, tidak mungkin mengirim data melalui media – media tersebut secara polos (*plain*), melainkan harus dilakukan proses pengamanan untuk data yang akan dikirim, salah satunya dengan cara melakukan enkripsi pada data tersebut. Metode pengamanan data yang digunakan adalah kriptografi, kriptografi merupakan salah satu pengamanan data yang dapat digunakan untuk menjaga kerahasiaan data, serta keaslian pengirim.

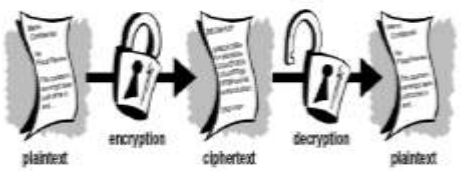
Dalam makalah ini akan diulas aplikasi kriptografi dengan metode vernam cipher dan metode pemutasi biner dengan tujuan meningkatkan keamanan data agar informasi yang bersifat rahasia di-enkripsi terlebih dahulu sebelum dikirim melalui internet agar tidak dapat di ketahui, di modifikasi atau dimanfaatkan oleh orang lain yang tidak bekepentingan.

Landasan Teori

Prinsip Dasar Kriptografi

Ilmu kriptografi adalah ilmu yang mempelajari tentang penyembunyian huruf atau tulisan sehingga membuat tulisan tersebut tidak dapat dibaca oleh orang yang tidak berkepentingan. Kriptografi sudah dipakai sejak jaman Julius Caesar dimana akan mengirimkan pesan kepada panglimanya tetapi tidak mempercayai kurir pembawa pesan tersebut. Kriptografi mempunyai 2 (dua) bagian yang penting, yaitu enkripsi dan dekripsi. Enkripsi adalah proses dari penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya. Dekripsi sendiri berarti merubah pesan yang sudah disandikan menjadi pesan aslinya. Pesan asli biasanya disebut *plaintext*, sedangkan pesan yang sudah disandikan disebut *ciphertext.*

Pada Gambar 1 dapat dilihat bahwa masukan berupa *plaintext* akan masuk ke dalam blok enkripsi dan keluarannya akan berupa *ciphertext*, kemudian *ciphertext* akan masuk ke dalam blok dekripsi dan keluarannya akan kembali menjadi *plaintext* semula.



Gambar.1 Proses Enkripsi dan Dekripsi

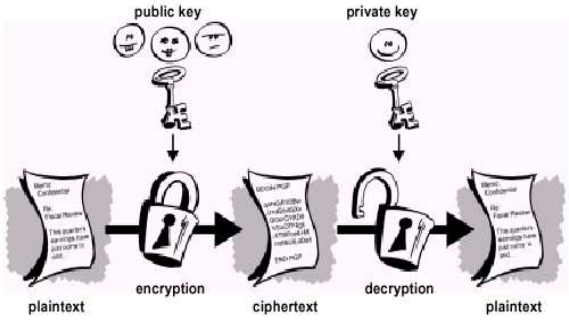
Ada 2 (dua) model algoritma enkripsi yang menggunakan kunci, yaitu kunci simetrik dan kunci asimetrik. Enkripsi kunci simetrik yang biasanya disebut enkripsi konvensional adalah enkripsi yang menggunakan kunci yang sama untuk enkripsi maupun dekripsi, dari Gambar 2 terlihat bahwa untuk mengenkripsi maupun mendekripsi pesan hanya menggunakan satu buah kunci (K) saja.

Gambar 2 Enkripsi-dekripsi Kunci Simetrik

Penggunaan metode ini membutuhkan persetujuan antara pengirim dan penerima tentang kunci sebelum mereka saling mengirim pesan. Keamanan dari kunci simetrik tergantung pada kerahasiaan kunci, apabila seorang penyusup dapat menemukan kunci maka dengan mudah dapat membaca pesan yang sudah dienkripsi. Enkripsi kunci simetrik dapat dibagi kedalam 2 (dua) kelompok yaitu metode *stream cipher* dan metode *block cipher*.

Enkripsi kunci asimetrik (biasa disebut enkripsi kunci publik) dibuat sedemikian rupa sehingga kunci yang dipakai untuk enkripsi berbeda dengan kunci yang dipakai untuk dekripsi. Enkripsi kunci *public* disebut demikian karena kunci untuk enkripsi boleh disebarluaskan kepada umum sedangkan kunci untuk mendekripsi hanya disimpan oleh orang yang bersangkutan.

Contohnya seperti pada Gambar 2.3 bila seseorang ingin mengirim pesan kepada orang lain maka orang tersebut menggunakan kunci *public* orang tersebut untuk mengenkripsi pesan yang kita kirim kepadanya lalu orang tersebut akan mendekripsi pesan tersebut dengan kunci privat miliknya.



Gambar 3 Enkripsi Kunci Asimetrik

Tujuan Kriptogtrafi

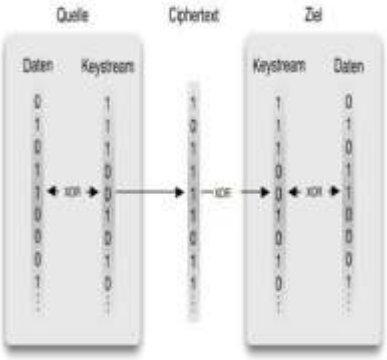
Seperti juga perkembangan ilmu kriptografi, tujuan-tujuan dari kriptografi teruslah berkembang. Bila pertama kali dibuat hanya untuk keamanan data saja, tetapi sekarang semakin banyak tujuan-tujuan yang ingin dicapai, yaitu:

1. *Privasi*, Musuh tidak dapat membongkar tulisan yang kita kirim.
2. *Autentikasi*, Penerima pesan dapat meyakinkan dirinya bahwa pesan yang diterima tidak terjadi perubahan dan berasal dari orang yang diinginkan.
3. *Tanda tangan*, penerima pesan dapat meyakinkan pihak ketiga bahwa pesan yang yang diterima berasal dari orang yang diinginkan.
4. *Minimal*, Tidak ada yang dapat berkomunikasi dengan pihak lain kecuali berkomunikasi dengan pihak yang diinginkan.
5. *Pertukaran bersama*, suatu nilai (misalnya tanda tangan sebuah kontrak) tidak akan dikeluarkan sebelum nilai lainnya (misalnya tanda tangan pihak lain) diterima.
6. *Koordinasi*, di dalam komunikasi dengan banyak pihak, setiap pihak dapat berkoordinasi untuk tujuan yang sama walaupun terdapat kehadiran musuh.

Stream Cipher

Stream cipher atau *stream encryption* merupakan suatu teknik enkripsi data dengan cara melakukan transformasi dari tiap *bit* secara terpisah berdasarkan posisi tiap bit dalam aliran data yang biasanya dikendalikan menggunakan operasi *XOR*. Enkripsi aliran data merupakan hasil dari operasi XOR antara setiap bit *plaintext* dengan setiap bit kuncinya. Pada *stream chiper* bila terjadi kesalahan selama transmisi maka kesalahan pada teks enkripsi penerima akan terjadi tepat di tempat kesalahan tersebut terjadi. Dalam praktek pertimbangan kesalahan yang mungkin terjadi

sangatlah penting untuk penentuan teknik enkripsi yang akan digunakan.



Gambar 4 Stream Cipher

Metode vernam cipher

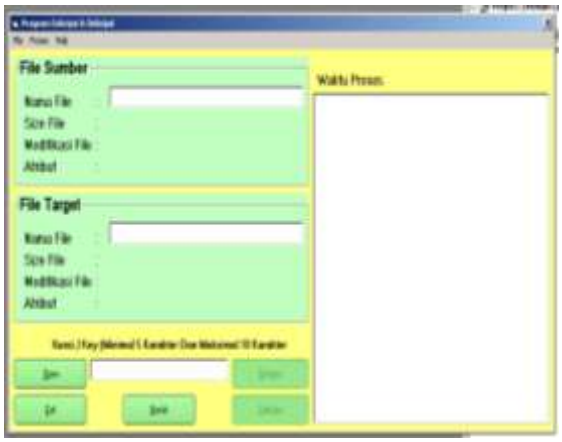
Kriptografi bagi kebanyakan orang adalah sesuatu yang sangat sulit dan kita sebagai pemula cenderung malas untuk mempelajarinya. Namun ada sebuah metode kriptografi yang agak mudah untuk dipelajari dan para ahlipun telah menyatakan bahwa metode ini merupakan metode kriptografi yang cukup aman untuk digunakan. Metode tersebut biasa dikenal dengan nama *One Time Pad* (OTP) atau yang lebih dikenal dengan sebutan *Vernam Cipher*. *Vernam Cipher* diciptakan oleh Mayor J. Maugborne dan G. Vernam pada tahun 1917.

Algoritma *One Time Pad* (OTP) merupakan algoritma berjenis *symetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara stream cipher yang berasal dari hasi *XOR* antara bit plaintext dan bit key. Pada metode ini plain text diubah kedalam kode ASCII dan kemudian dikenakan operasi *XOR* terhadap kunci yang sudah diubah ke dalam kode ASCII.

PEMBAHASAN

Proses Enkripsi Data

Enkripsi data merupakan bagian awal dari proses pengamanan data. Dalam proses enkripsi ini data ang asli akan dilakukan proses pengacakan dengan algoritma yang sudah ditentukan. Tampilan awal dari aplikasi yang dikembangkan seperti pada gambar 7.



Gambar 5 Tampilan Enkripsi Data

Tampilan ini meminta pemakai untuk memasukan nama file ang akan dienkripsi. Adapun langkah-langkah proses enkripsi adalah :

1. Proses *enkripsi* dilakukan setelah menginputkan *file* yang akan dienkripsi dan menginput kunci / *key* pada *text key*.
2. Form *enkripsi* data ditampilkan setelah tombol *open*
3. Frame *file* sumber berfungsi untuk menampilkan informasi dari *file file* sumber / asal yang akan dilakukan *enkripsi*.
4. Frame *file* target berfungsi untuk menampilkan informasi dari *file file* target / tujuan yang akan telah dilakukan *enkripsi*.
5. Tombol open berfungsi untuk mencari dan membuka *file* yang akan di*enkripsi*
6. Tombol *enkripsi* berfungsi untuk melakukan proses perubahan / enkripsi dari *plaintext* menjadi *ciphertext.*
7. Proses *enkripsi* dilakukan setelah menginputkan type *file* yang akan dienkripsi dan menginput kunci / *key* pada *text key*.
8. Tombol *exit* berfungsi untuk keluar dari *form.*
9. Waktu proses berfungsi untuk menampilkan durasi waktu yang dibutuhkan untuk melakukan proses

Kode program enkripsi adalah sebagai berikut :

```
Open TxtSumber.Text For Binary As 2
Put 1, , "Day" & Chr$(0) & "at"
Put 1, , PNam$
Put 1, , Nam$
Put 1, , Atri$
Put 1, , PTang$
Put 1, , Tang$
Put 1, , PHKey$
Put 1, , HKey$
Mulai = Now
Prog.Max = LOF(2)
K = LOF(2) \ 1024
If K Then
A$ = Space$(1024)
For L = 1 To K
Get 2, , A$
B$ = Enkrip$(A$, Key1$, Key2$)
Put 1, , B$
Prog.Value = Prog.Value + 1024
Next
End If
K = LOF(2) Mod 1024
If K Then
A$ = Space$(K)
Get 2, , A$
B$ = Enkrip$(A$, Key1$, Key2$)
Put 1, , B$
Prog.Value = Prog.Value + K
```

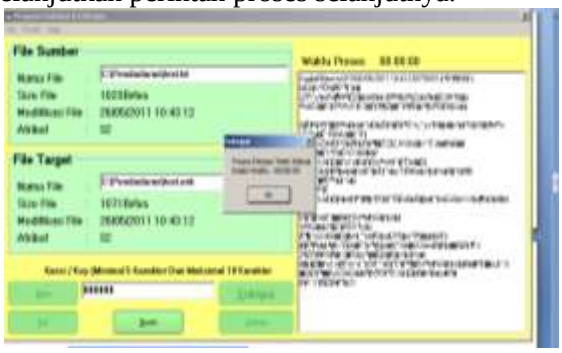
Setelah ditentukan jenis *file* maka akan terlihat informasi atau isi dari *file* tersebut. (gambar 6)

Gambar 6 Tampilan Informasi file sumber

Sebelum melakukan enkripsi atau dekripsi *file* yang perlu dilakukan adalah memasukan kunci atau key dan harus diingat karakter yang diinputkan karena setiap karakter mempunya kode ASCII yang berbeda.

Gambar 7 Tampilan Input Kunci / Key

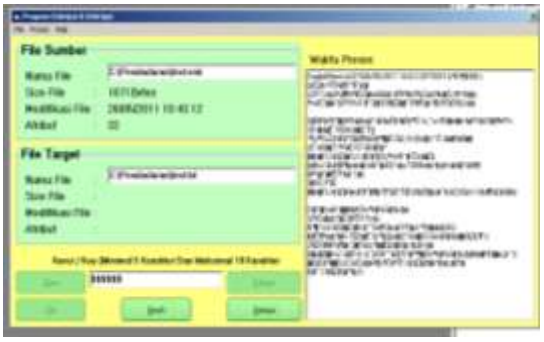
Setelah kunci diinputkan maka user harus menekan tombol enkripsi program akan mengecek apakah kunci sama dengan kosong, kunci kurang dari lima karakter, lebih dari lima karakter atau kurang lebih dari sepuluh karakter. Jika kunci memenuhi syarat maka program akan melanjutkan perintah proses selanjutnya.



Gambar 8 Tampilan Proses Enkripsi

Proses Dekripsi

Proses dekripsi merupakan proses yang dilakukan untuk mengembalikan *file* dari bentuk simbol-simbol kembali ke bentuk semula .Bentuk tampilan dekripsi data seperti pada gambar dibawah ini :



Gambar 9. Tampilan Dekripsi Data

Ketentuan dekripsi data :

1. Form *dekripsi* data ditampilkan setelah tombol open diklik
2. Frame *file* sumber berfungsi untuk menampilkan informasi dari *file file* sumber / asal yang akan dilakukan *dekripsi*.
3. Frame *file* target berfungsi untuk menampilkan informasi dari *file file* target / tujuan yang akan telah dilakukan *dekripsi*.
4. Tombol open berfungsi untuk mencari dan membuka *file* yang akan di *dekripsi*
5. Tombol *dekripsi* berfungsi untuk melakukan proses perubahan / *dekripsi* dari *ciphertext* menjadi *plaintext.*
6. Proses *dekripsi* dilakukan setelah menginputkan type *file* yang akan dienkripsi dan menginput kunci / *key* pada *text key*.

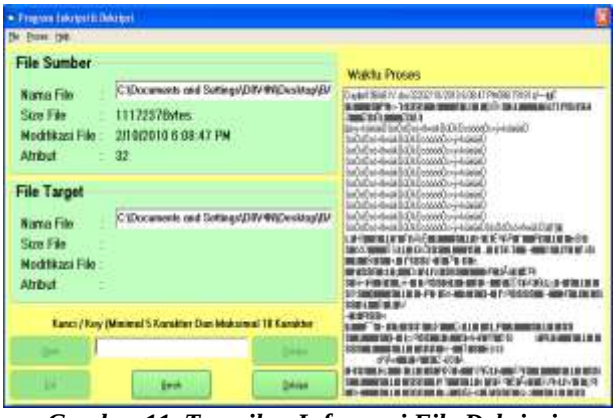
Kode program untuk proses dekripsi adalah sebagai berikut :

```
Open TxtSumber.Text For Binary As 2
   H1$ = Space$(6)
   Get 2, 1, H1$
   H2$ = Space$(2)
   Get 2, , H2$
   H3$ = Space$(Val(H2$))
   Get 2, , H3$
   H4$ = Space$(2)
   Get 2, , H4$
   H5$ = Space$(2)
   Get 2, , H5$
   H6$ = Space$(Val(H5$))
   Get 2, , H6$
   H7$ = Space$(2)
   Get 2, , H7$
   H8$ = Space$(Val(H7$))
   Get 2, , H8$
If H8$ <> HeaderKey$(TxtKey.Text) Then
   MsgBox " Key Anda Salah ", vbOKOnly + vbCritical,
"Perhatian"
   Close 2
   TxtKey.Locked = False
   TxtKey.Text = ""
   TxtKey.SetFocus
   'XpExit.Enabled = True
```

Jika ingin menentukan isi fila yang akan di didekripsi cukup klik tombol open pada button atau menu bar, maka akan tampil menu open *file* selanjutnya menentukan jenis *file* yang akan dienkripsi atau dekripsi.

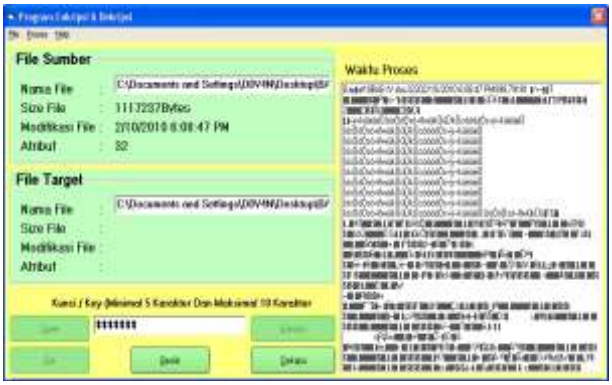
Gambar 10 Tampilan Open File Dekripsi

Setelah ditentukan jenis *file* maka akan terlihan informasi dari *file* tersebut dan pada rtfbox akan tampil isi dari yang akan didekripsi.



Gambar 11. Tampilan Informasi File Dekripsi

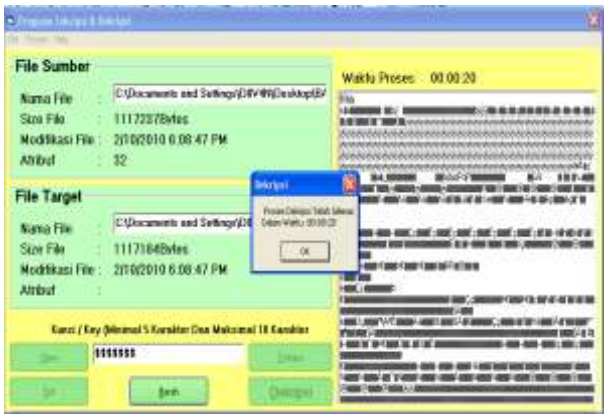
Sebelum melakukan enkripsi atau dekripsi *file* yang perlu dilakukan adalah memasukan kunci atau key dan harus diingat karakter yang diinputkan karena setiap karakter mempunya kode ASCII yang berbeda.



Gambar 12. Tampilan Input Key / Kunci File Dekripsi

Setelah kunci diinputkan maka user harus menekan tombol enkripsi program akan mengecek apakah kunci sama dengan kosong, kunci kurang dari lima karakter, lebih dari lima karakter atau kurang lebih dari sepuluh karakter.

Jika kunci memenuhi syarat maka program akan melanjutkan perintah proses selanjutnya.



Gambar 13. Tampilan Hasil Proses Dekripsi

Kesimpulan

Hasil desain aplikasi dan penyusunan aplikasi enkripsi dan dekripsi dengan metode vernam cipher dan permutasi biner beserta tahapan implementasi yang dilakukan diperoleh kesimpulan berikut :

1. Aplikasi ini menggunakan dua metode enkripsi dan dekripsi agar lebih aman dan terjamin kerahasiaan data.
2. Menggunakan *header* untuk menyimpan informasi-informasi seperti nama *file*, atribut *file* dan tanggal *file* sebelum dienkripsi, supaya pada saat didekripsi nama *file*, atribut *file* dan tanggal *file* tidak berubah.
3. Aplikasi enkripsi dan dekripsi diberi kunci agar tidak sembarang user dapat mendekripsi file
4. Waktu yang dibutuhkan untuk proses enkripsi dan dekripsi dipengaruhi oleh kecepatan komputer yang digunakan dan ukuran file.

5.1. Saran

Meskipun perancangan program enkripsi, dekripsi dan tahapan implementasi telah memenuhi kebutuhan proses pengamanan data namum aplikasi perlu dilakukan pengembangan agar lebih sempurna lagi sebagai berikut :

1. User dapat melihat kunci yang dimasukan pada proses enkripsi pada *header* kunci yang telah dienkripsikan terlebih dahulu, supaya jika user lupa kunci yang dimasukan pada saat enkripsi, maka *user* dapat mengetahui kunci untuk melakukan dekripsi.
2. Perlu diubah algoritmanya agar proses enkripsi dan dekripsi menjadi lebih cepat.
3. Program ini akan di lakukan ujicoba pada Dinas Pendidikan Dan Pengajaran Provinsi Papua.

Daftar pustaka

Bambang Riadi, *Penentuan Nilai Permutasi Dengan Lexicographic Order.* http://bambangriadi.com/br/tag/permutasi/

Burton Rosenberg , *Vernam Cipher, a perfect cipher.* http://www.cs.miami.edu/~burt/learning/Csc609.051/notes/02.html

Crasher. 2006. *Algoritma Enkripsi One Time Pad*. http://www.code_attack.com.

One Time Pad. http://www.topsecretcripto.com

Jarecki, Stanilw, (2004), Introduction to Cryptography : Lecture 1 : *Crypto Overview, Perfect Secrecy, One Time Pad*.

Rachmad, *Teknik Dasar Kriptografi.* http://blog.re.or.id/teknik-dasar-kriptografi-permutasi.htm

Wagner, Neal R., (2002), *The Laws of Cryptography : Perfect Cryptography : The One-Time Pad.* http://www.cs.utsa.edu/~wagner/laws/pas.html

Wahana Komputer. 2003. *Memahami Model Enkripsi dan Security Data*. Yogyakarta: Andi Offset.