# IMPLEMENTASI ALGORITMA MERKLE-HELLMAN ADDITIVE KNAPSACK

**Haeni Budiati, M.kom**
Ilmu Komputer, FMIPA, UKRIM

***Abstrak***

*Perkembangan komputer dan teknik komunikasi data menjadikan kriptografi juga memiliki peran penting dalam kegiatan pengiriman pesan atau transaksi bisnis. Ketersediaan (mencegah kegagalan akses oleh orang yang tidak berhak). kriptografi yaitu teknik kunci rahasia dan teknik kunci publik. Dalam teknik kunci rahasia, yang disebut juga sistem kriptografi simetris, dua belah pihak yang akan saling berkomunikasi harus memiliki kunci yang sama.*

*Kata kunci : Kriptografi, polyalphabetic.*

## 1. PENDAHULUAN

### 1.1. Latar Belakang Masalah

Pada awalnya Kriptografi hanya digunakan oleh kalangan militer, karena merekalah yang banyak berkecimpung dengan informasi rahasia. Perkembangan komputer dan teknik komunikasi data menjadikan kriptografi juga memiliki peran penting dalam kegiatan pengiriman pesan atau transaksi bisnis.

Dalam era globaliasi ini seolah-olah tidak ada lagi batas antar negara terlebih dengan hadirnya internet, para pengguna di seluruh dunia, dapat memanfaatkan semua fasilitas yang tersedia di dalamnya. Terdapat 3 faktor keamanan data yang harus mendapat perlindungan yaitu : Kerahasiaan (mencegah diketahui oleh orang yang tidak berhak), Integritas (mencegah diubah oleh orang yang tidak berhak), Ketersediaan (mencegah kegagalan akses oleh orang yang tidak berhak). Kriptografi menjadi solusi yang sangat penting dalam menjaga keamanan data.

### 1.2 Tujuan dan Manfaat

Tujuan penulisan dari Tugas Akhir ini adalah:

a) untuk memahami cara kerja program enkripsi-dekripsi, melalui penerapan metode Additive Knapsack,

b) membuat program pengamanan data teknik kunci publik, dengan cara menyandikan data sedemikian rupa kemudian menyimpannya dalam bentuk berkas dan dapat mengembalikan ke bentuk semula

## 2. LANDASAN TEORI

### 2.1. Kriptografi

Kriptografi (*cryptography*) diambil dari bahasa Yunani yang secara harafiah mempunyai arti penulisan kalimat rahasia (sandi). Kriptografi adalah ilmu untuk menjaga kerahasiaan data dengan menerapkan rumus matematika tertentu. Kriptografi memiliki dua proses yaitu enkripsi dan dekripsi. Enkripsi adalah proses mengubah sebuah pesan asli (*plaintext*) dengan rumus tertentu agar menjadi pesan sandi (*ciphertext* atau disebut juga *cryptogram*) sedangkan dekripsi, yaitu suatu proses transformasi dari informasi yang telah *diencrypt* (disandikan dan diacak) kembali menjadi bentuk aslinya.

### 2.4. Sistem Kriptografi

Sistem kriptografi adalah fasilitas penyandian untuk mengkonversi *plaintext* menjadi *ciphertext* yang meliputi lima komponen yaitu data plaintext, data ciphertext, kunci kriptografi, fungsi transformasi enkripsi dan fungsi transformasi dekripsi. Secara umum, operasi enkripsi dan dekripsi dapat dijelaskan oleh beberapa fungsi dibawah ini :

$$Y = E_{ke}(X) \quad \text{(enkripsi)}$$

$$X = D_{kd}(Y) \quad \text{(dekripsi)}$$

Dimana X adalah data asli (plaintext) dan Y adalah data yang dikaburkan (ciphertext), sedangkan $k_e$ adalah kunci enkripsi dan $k_d$ adalah kunci dekripsi.

### 2.3. Metode-Metode Kriptografi

#### 2.3.1. Metode Transposisi

Transposition ciphers adalah blok cipher yang menggunakan permutasi yang sederhana,dengan mengatur ulang posisi simbol-simbol dalam blok. Pada proses enkripsi: $Y = E_{ke}(X)$ dan pada proses dekripsi: $X = D_{kd}(Y)$.

### 2.3.2. Metode Substitusi

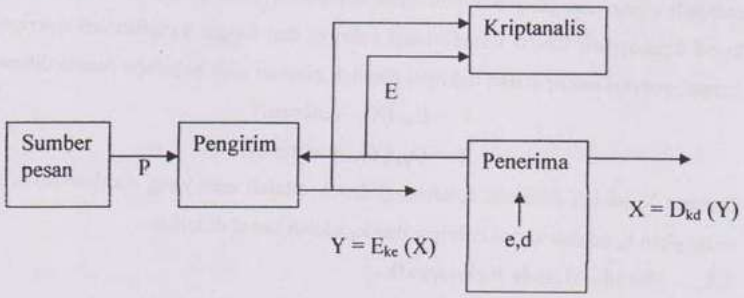
Ada empat jenis metode subtitusi yaitu : *monoalphabetic, homophonic, polyalphabetic dan polygram*. Penyandian substitusi yang sederhana dinamakan *monoalphabetic* karena adanya korespodensi satu-satu antara sebuah karakter dan karakter penggantinya. Teknik penyandian substitusi *homophonik* memetakan setiap karakter x dalam plaintext menjadi sekelompok elemen ciphertext f(x) yang disebut homophones. Pemetaan f dari plaintext menjadi ciphertext adalah satu-kesatu. Teknik penyandian substitusi *polyalphabetic* adalah sebuah teknik penyandian substitusi berganda yang melibatkan pemakaian kunci-kunci yang berbeda.

### 2.3.3. Metode Product Ciphers

Caesar Cipher (ditemukan oleh Julius Caesar) adalah salah satu contoh pemakaian teknik penyandian substitusi, yaitu substitusi *monoalphabetic*. Kunci tetap dengan cara huruf ke-i diganti dengan huruf ke (i+k) modulo 26.

### 2.3.4. Teknik Kriptografi

Terdapat dua teknik kriptografi yaitu teknik kunci rahasia dan teknik kunci publik. Dalam teknik kunci rahasia, yang disebut juga sistem kriptografi simetris, dua belah pihak yang akan saling berkomunikasi harus memiliki kunci yang sama. Masalah yang muncul apabila kedua belah pihak berada dikondisi yang berjauhan adalah bagaimana cara mengirim kunci (yang tidak mungkin dienkripsikan) kepihak kedua dengan aman. Kerawanan akan makin bertambah apabila komunikasi harus dilakukan oleh lebih dari dua pihak.




*Gambar 2.3.4. Tinjauan matematis sistem kriptografi kunci public*

## 2.4. Sistem Kriptografi Merkle-Hellman Knapsack

Konsep publik-key cryptosystem diperkenalkan oleh Whitfiel Diffie dan Martin Hellman pada tahun 1976. Pada tahun 1978, Merkle-Hellman mengusulkan sistem kriptografi kunci publik M-H berdasarkan *trapdoor knapsack problem*. Sebuah vektor knapsack sederhana $K'_p = (k'_{p1}, k'_{p2}, \ldots k'_{pn})$ dimana $k'_{pi}$ $(1<=i<=n)$ adalah integer, merupakan sebuah vektor *supericreasing*, artinya setiap elemen vektor tersebut mempunyai sifat lebih besar daripada jumlahan elemen-elemen sebelumnya, perancang memilih dua integer besar m dan w yang m>w dan gcd (m,w)=1, yang artinya m dan w relatif prima. Selain itu, m harus sebuah integer yang lebih besar dari $\sum k'_i (1 \leq i \leq n)$, jadi $m > \sum k'_i$. Dengan demikian vektor knapsack trapdoor $k_p$ dihitung dari vector knapsack sederhana $k'_p$ melalui v yang merupakan invers multiplikatif dari w modulo m sehingga $vw \equiv 1 \pmod m$, dimana $v = w^{-1}$. Jadi sebuah trapdoor knapsack $k_p$ dihasilkan dengan mengalikan setiap komponen dari vektor knapsack sederhana $k'_p$ dengan w sehingga

$K_i \equiv wk'_i \pmod m$

$K_p \equiv wk'_p \pmod m$

### 2.4.1. Metode Additive Knapsack

Metode additive knapsack adalah sebuah bentuk penyelesaian masalah pemulihan data asli dari data sandi yang paling sederhana, dengan salah satu ciri khas yaitu kunci rahasia yang digunakan untuk menciptakan kunci publik harus *superincreasing*, artinya setiap elemen vektor tersebut mempunyai sifat lebih besar daripada jumlahan elemen-elemen sebelumnya.

Asumsikan sebuah kunci $K = (k_1, k_2, k_3, \ldots, k_n)$ dengan $K_i$ adalah integer $1 < i < n$ dan suatu n-bit *plaintext yaitu* $X = (x_1, x_2, x_3, \ldots, x_n)$ dimana $x_i$ adalah sebuah bit biner untuk $1 \leq i \leq n$. Setiap kriptografi *knapsack* mengenkripsikan n-bit *plaintext* menjadi m-bit *ciphertext*, dimana n<m, seperti yang dirumuskan dalam formula berikut : $Y = K.X = k_1.x_1 + k_2.x_2 + k_3.x_3 + \ldots + k_n.x_n = \sum k_i.x_i$. Pemulihan data asli X dari Y dan K melibatkan pemecahan masalah knapsack dan umumnya akan sulit bila n sangat besar dan kunci K dipilih secara acak dan harus memenuhi kriteria *superincreasing*.

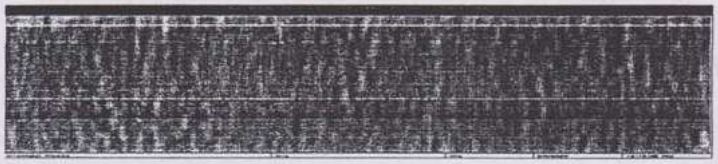
# 3. HASIL DAN PEMBAHASAN PROGRAM

## 3.1. Hasil

Sesuai dengan rancangan program yang telah dijelaskan di bab sebelumnya, bahwa tampilan yang dibuat terdiri dari 7 form dan setelah program dijalankan, tampilan yang dapat dilihat adalah tampilan tampilan menu utama, tampilan untuk file baru, tampilan untuk buka file, tampilan untuk buka hasil, tampilan untuk proses enkripsi, tampilan untuk proses dekripsi, tampilan untuk proses perbandingan.

### 3.1.1. Form Menu Utama

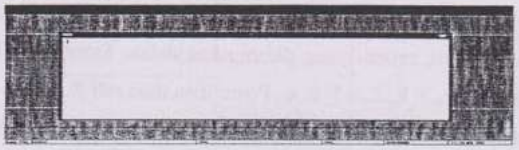
Pada tampilan form menu utama, terdapat beberapa pilihan menu file, antara lain: File, Proses, Jendela, Selesai. Jika user memilih *File* maka akan ditampilkan pilihan antara lain: Baru. Buka file dan Simpan, pada *Proses* akan ditampilkan pilihan antara lain: Enkripsi, Dekripsi, Perbandingan, tabel karakter, tabel ascii dan tabel ascii encode, menu *Jendela* memberikan pilihan-pilihan kepada user untuk mengatur posisi jendela yang telah dibuka. User bisa memilih tombol *Selesai* jika ingin keluar dari program. Gambar 4.3 adalah merupakan tampilan menu utama.



*Gambar 3.1.1 Tampilan Form Menu Utama*

### 3.1.2. Form Baru

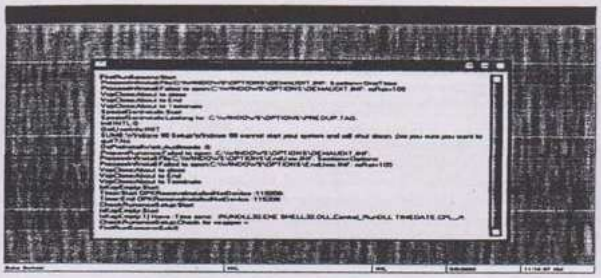
Form ini muncul setelah user menekan file baru pada pilihan menu utama, selanjutnya user dapat menuliskan sebuah file dalam bentuk text ke dalam form tersebut yang kemudian dapat disimpan untuk dapat diproses.



*Gambar 3.1.2 Tampilan Form Baru*

### 3.1.3. Form Buka File

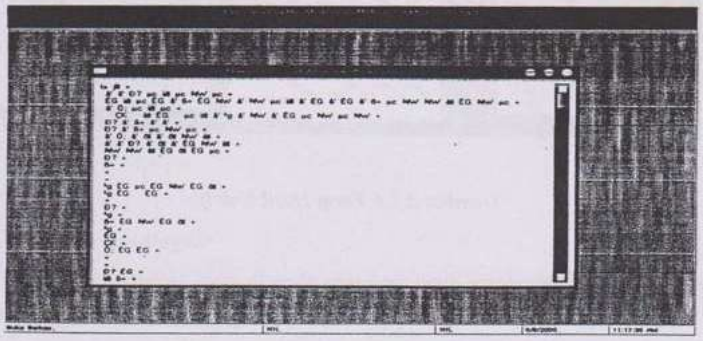
Form buka file muncul pada saat user menekan file buka pada pilihan menu utama. Form ini menampilkan file yang sudah ada di dalam berkas yang kemudian dapat di simpan dalam bentuk text.



*Gambar 3.1.3 Tampilan Form Buka File*

### 3.1.4. Form Buka Hasil

Form ini menampung hasil yang diperoleh dari proses yang sudah dijalankan. Hasil proses enkripsi disimpan dengan nama **txt*, sedangkan hasil untuk proses dekripsi disimpan dengan nama **enc*.

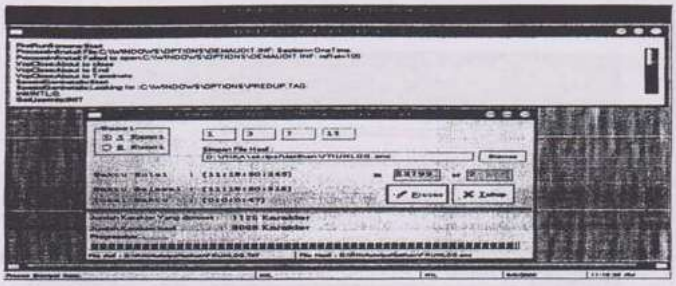


*Gambar 3.1.4 Tampilan Form Buka Hasil*

### 3.1.5. Form Enkripsi

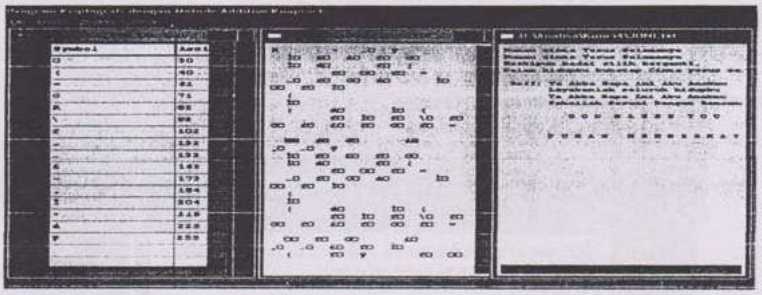
Dari *form* menu utama kita pilih menu *proses*, kemudian akan muncul menu enkripsi, dekripsi, perbandingan,tabel karakter, tabel ascii dan tabel ascii encode. Pada menu-menu tersebut kita memilih menu enkripsi dan akan muncul *form* untuk pengisian

*kunci* yang akan dienkripsi ada dua pilihan kunci yaitu 4 dan 8, simpan file hasil, hasil nilai m dan w yang random, waktu mulai, waktu selesai, total waktu, jumlah karakter yang diproses dan jumlah karakter hasil . *Form* tersebut dapat dilihat seperti gambar di bawah ini:



*Gambar 3.1.5 Tampilan Form Enkripsi*

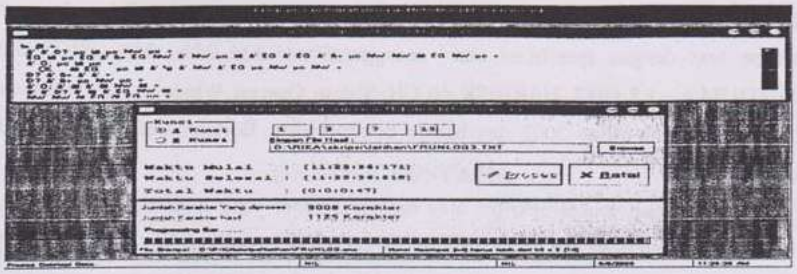
Dibawah ini ditampilkan form jendela hasil proses enkripsi.



*Gambar 3.1.6 Form Hasil Enkripsi*

### 3.1.6. Form Dekripsi

Dari *form* menu utama kita pilih menu *proses*, kemudian akan muncul menu enkripsi, dekripsi dan perbandingan. Pada menu-menu tersebut kita memilih menu dekripsi dan akan muncul *form* untuk pengisian *kunci* yang akan didekripsi ada dua pilihan kunci yaitu 4 dan 8, simpan file hasil, waktu mulai, waktu selesai, total waktu, jumlah karakter yang diproses dan jumlah karakter hasil . *Form* tersebut dapat dilihat seperti gambar di bawah ini:



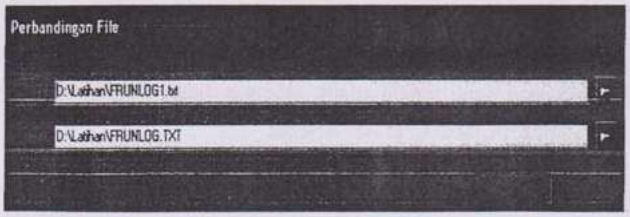
*Gambar 3.1.7 Tampilan Form Dekripsi*

Dibawah ini ditampilkan form jendela hasil proses dekripsi :

*Gambar 3.1.8 Form Hasil Dekripsi*

### 3.1.7. Form Perbandingan

Form perbandingan muncul pada saat user ingin menbandingkan file, antara file asli dengan file hasil dekripsi.




*Gambar 3.1.9 Form Perbandingan*

## 4. PEMBAHASAN

Pada bagian ini akan dilakukan uji coba untuk enkripsi dan deskripsi data yang bertipe text dengan spesifikasi alat menggunakan RAM 256 MB, PROSESOR PENTIUM 4 ; 1,8 GHZ, HARDISK 40 GB, Sistem Operasi Windows XP Profesional, Office Microsoft office 2003 dan bahasa pemrograman Borland Delphi 7.0. Dengan ketentuan :

1. Dari setiap proses enkripsi atau deskripsi kunci yang digunakan adalah sama, proses enkripsi dan deskripsi kunci 4 mengambil contoh kunci adalah 5,8,15,30 dengan m= 80 dan w= 3. Sedangkan dengan kunci 8 mengambil contoh kunci 5,8,15,30,70,130,380,680 dengan m= 2386 dan w=215.
2. File yang akan diproses merupakan file yang memiliki isi dengan variasi data berpola dan acak. Data berpola diambil contoh yaitu: a,aba,abc,abcd,abcde.
3. Ukuran data 10,100,1000,10000,100000 karakter.

## 5. KESIMPULAN

Dari berbagai uraian pada bab sebelumnya, maka didapat beberapa kesimpulan sebagai berikut:

1. Waktu pada proses enkripsi dan dekripsi dipengaruhi oleh beberapa hal diantaranya adalah ukuran file dan jumlah kunci. Semakin besar ukuran file yang diproses, maka waktu yang dihasilkan akan lebih lama.dan jumlah kunci yang digunakan lebih besar maka waktu proses dihasilkan juga akan semakin lama.
2. Waktu yang digunakan untuk proses enkripsi lebih lama daripada waktu untuk proses dekripsi dan waktu pada proses enkripsi dan dekripsi tidak dipengaruhi oleh bentuk pola huruf.
3. Keefektifan dari algoritma ini dapat dilihat dari cara kerja yang digunakan proses enkripsi dan dekripsi sangat sederhana, tidak membutuhkan banyak perhitungan yang terlalu rumit, sehingga tidak membutuhkan waktu yang lama untuk melakukan proses tersebut.

4. Jumlah panjang karakter hasil enkripsi lebih banyak dari jumlah panjang karakter asli yang telah diproses.

## DAFTAR PUSTAKA

Denning,Dorothy Elizabeth R., *Cryptography And Data Security*, Addison – Wesley Publishing Compani, 1983.

Diffle,W.,and Hellman,MF., *Exhautive Cryptonalys Of the NBS Data Encryption Standart*, IEEE Computer Magazine, 1997.

Hellman, Martin E., *The Mathematics Of Public Key Cryptography*, Scientific American, 1983.

Man Young Rhee., *Cryptography And Data Secure Communications*, McGraw-Hill Book Co, 1994.

William Stallings, Ph.D., *Network And Internet Work Principles And Practice Computer Network*, Prentice-Hall, 1995.

Yusuf Kurniawan.,MT., *Kriptografi Keamanan Internet Dan Jaringan Komunikasi*, Informatika Bandung, 2004.