Rancang Bangun Aplikasi Enkripsi dan Dekripsi Email Dengan Menggunakan Algoritma Advanced Encryption Standard Dan Knapsack

Timothy John Pattiasina, ST., M.Kom.
Jurusan Teknik Informatika
Institut Informatika Indonesia
Jl. Raya Sukomanunggal Jaya 3, Surabaya
temmy@ikado.ac.id

ABSTRAK

Advanced Encryption Standard (AES) dan *Knapsack* adalah dua algoritma enkripsi simetris dan asimetris yang paling sering digunakan. Penelitian ini menganalisa kedua algoritma *AES* dan algoritma *Knapsack*. Prototipe aplikasi enkripsi *email* ini dirancang dengan menggabungkan karateristik algoritma *AES* dan *Knapsack* untuk memecahkan masalah keamanan *email*. Algoritma *AES* digunakan untuk mengenkripsi dan deskripsi *email* berupa teks atau file, sedangkan Algoritma *Knapsack* di gunakan untuk mengenkripsi kunci *AES*. Enkripsi hybrid yang diterapkan pada aplikasi bertujuan untuk menambah keamanan informasi dalam sebuah jaringan.

Kata Kunci : Advanced Encryption Standard, *Knapsack*, Hybrid Enkripsi, Dekripsi

1. PENDAHULUAN

Seiring dengan perkembangan yang terjadi pada teknologi komputer saat ini, pertukaran informasi sangat dibutuhkan. Pengiriman informasi melalui jaringan elektronik, khususnya *email* memerlukan suatu proses yang menjamin keamanan dan keutuhan dari informasi yang dikirimkan.

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu pesan, data, atau informasi. Dalam hal ini sangat terkait dengan betapa pentingnya pesan, data, atau informasi tersebut dikirim dan diterima oleh pihak atau orang yang berkepentingan, apakah pesan, data, atau informasi masih authenticity. Pesan, data, atau informasi akan tidak berguna lagi apabila ditengah jalan informasi itu disadap atau dibajak oleh orang yang tidak berhak atau berkepentingan.

Untuk menjaga keamanan dan kerahasiaan pesan, data, atau informasi dalam suatu jaringan komputer maka diperlukan beberapa enkripsi guna membuat pesan, data, atau informasi tersebut tidak dapat dibaca atau dimengerti oleh sembarang orang, kecuali untuk penerima yang berhak, oleh sebab itu penulis membuat sebuah aplikasi enkripsi dan dekripsi *email* dengan menggunakan algoritma *Advanced Encryption Standard* (*AES*) dan *Knapsack*, agar data atau informasi tersebut masih terjaga keamanannya.

2. METODOLOGI PENELITIAN

Dalam penelitian ini dijabarkan mengenai tinjauan pustaka yang menjadi dasar dilakukannya penelitian ini. Dimana seluruh tinjauan pustaka mencakup semua hal terkait dengan penelitian yang dilakukan.

A. Kriptografi

Kriptografi (cryptography) berasal dari bahasa Yunani yang terdiri dari dua suku kata yaitu kryptos yang artinya tersembunyi dan graphein yang artinya tulisan. Jadi kata kriptografi dapat diartikan sebagai frase tulisan tersembunyi [1]. Kriptografi merupakan studi teknik matematika yang berhubungan dengan keamanan informasi seperti kerahasiaan, integritas data, autentikasi. Teknik ini digunakan untuk mengubah data ke dalam kode-kode tertentu, dengan tujuan informasi yang disimpan atau ditransmisikan melalui jaringan yang tidak aman. Istilah yang digunakan dalam kriptografi untuk melakukan proses kerjanya adalah sebagai berikut:

a. *Plaintext*

   Plaintext merupakan pesan asli yang belum disandikan atau informasi yang ingin dikirimkan atau dijaga keamanannya.

b. *Ciphertext*

   Ciphertext merupakan pesan yang telah disandikan sehingga siap untuk dikirimkan.

c. Enkripsi

   Enkripsi merupakan proses yang dilakukan untuk menyandikan *Plaintext* menjadi *Ciphertext* dengan tujuan pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang.

d. Dekripsi

   Dekripsi merupakan proses yang dilakukan untuk memperoleh kembali *Plaintext* dari *Ciphertext*.

e. Kriptosistem

   Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

B. Algoritma Advanced Encryption Standard (AES)

Karena DES dianggap sudah tidak aman lagi, Agensi Departemen Perdagangan AS, National Institute of Standard and Technology yang sebelum tahun 1988 juga dikenal sebagai National Bureau of Standard, mengusulkan kepada Pemerintah Federal AS untuk merancang sebuah standard kriptografi baru.

Untuk menghindari kontroversi mengenai standard yang baru tersebut, sebagaimana terjadi pada pembuatan DES, dimana waktu itu NSA yang berperan sebagai penilai kekuatan algoritma, maka NIST mengadakan sayembara terbuka untuk membuat standard algoritma kriptografi yang baru sebagai pengganti DES. Standard tersebut nanti akan diberi nama *Advanced Encryption Standard* [2].

- Enkripsi Algoritma *AES*
  a. Mengekspansi kunci (Key Expansion)

     Kunci hasil ekspansi ini disebut dengan RoundKey yang kemudian digunakan pada tiap-tiap putaran transformasi.

  b. Melakukan penjumlahan bit antara block *Plaintext* dengan kunci yang terekspansi.

  c. Melakukan transformasi putaran sebanyak Nr kali sebagai berikut:

     1. *SubByte*

        Proses mensubstitusi *Plaintext* yang telah diekspansi ke dalam S-Box.

     2. *ShiftRows*

        Rotasi yang dilakukan mulai baris kedua hingga baris ke-4 ke kanan.

     3. *MixColumns* (untuk putaran ke Nr langkah ini tidak dilakukan) *State* yang dihasilkan dari proses *ShiftRows* di-XOR-kan dengan matrik yang telah ditentukan.

4. *AddRoundKey*

   Hasil dari *MixColumns* di-XOR-kan dengan RoundKey masing-masing putaran. RoundKey diperoleh pada proses ekspansi kunci.

- Dekripsi Algoritma *AES*

1. Tranformasi *Inverse SubByte*

   Tranformasi inverse *SubByte* merupakan kebalikan operasi subtitusi non-linear pada tiap-tiap byte dalam *state* dengan menggunakan tabel subtitusi yang dinamakan S-Box.

2. Tranformasi *Inverse* SiftRows

   Tranformasi *Inverse* SiftRows adalah menggeser kembali ke tempat semula isi baris 1,2 dan 3 dari *state* dengan jumlah pergeseran yang bervariasi.

3. Tranformasi *Inverse MixColumns*

   Tranformasi *Inverse MixColumns* merupakan perkalian terhadap *Inverse* matrik konstan yang dioperasikan kedalam kolom dalam *state*.

4. Tranformasi *Inverse MixColumns*

   Tranformasi *Inverse AddRoundKey* dengan cara menambahkan *Inverse* kunci rond ke *state* dalam operasi XOR.

C. Algoritma Knapsack

Knapsack merupakan optimasi pengangkutan barang atau disebut juga optimasi kombinatorial. *Knapsack* problem adalah suatu masalah bagaimana cara menentukan pemilihan barang dari sekumpulan barang di mana setiap barang tersebut mempunyai berat dan profit masing – masing, sehingga dari pemilihan barang tersebut didapatkan profit yang maksimum.

Tujuan *Knapsack* problem adalah agar mendapatkan keuntungan yang maksimum dari pemilihan barang tanpa melebihi kapasitas daya tampung media transportasi tersebut. Dalam teori algoritma, persoalan *Knapsack* termasuk ke dalam kelompok NP-complete. Persoalan yang termasuk NP-complete tidak dapat dipecahkan dalam orde waktu polynomial [5].

- Enkripsi Algoritma *Knapsack*

1. Menggunakan kunci publik untuk melakukan enkripsi.
2. *Plaintext* dipecah menjadi block bit yang panjangnya sama dengan kardinalitas barisan kunci publik.

3. Kalikan setiap bit di dalam block dengan elemen yang berkoresponden di dalam kunci public .

- Dekripsi Algoritma *Knapsack*
  1. Menggunakan kunci rahasia untuk melakukan dekripsi.
  2. Menghitung nilai n–1, yaitu kebalikan n modulo m, sedemikian sehingga n × n–1 º 1 (mod m).
  3. Mengalikan setiap kriptogram dengan n–1 mod m, lalu nyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci rahasia untuk memperoleh *Plaintext* dengan menggunakan algoritma pencarian solusi superincreasing *Knapsack* [3].

D. Microsoft Visual Studio

Microsoft Visual Studio merupakan salah satu bahasa pemrograman yang bisa membangun aplikasi-aplikasi di platform Microsoft .Net [4]. Dengan adanya Visual Studio, para programmer dapat membangun aplikasi Windows Form, Aplikasi web berbasi ASP, dan juga aplikasi command-line. Bahasa Visual Basic .NET menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic .NET versi sebelumnya yang diimplementasikan di atas .NET Framework. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu [4].

Microsoft Visual Basic.NET memiliki banyak fasilitas baru dan ditingkatkan seperti inheritance, interface, dan overloading yang menjadikannya sebagai bahasa pemrograman berorientasi objek yang tangguh. *Object Oriented Programming* merupakan kumpulan objek yang saling berinteraksi satu dengan lainnya. OOP akan mendekomposisikan masalah dunia nyata dan dinamakan *class* ataupun type[6].

3. HASIL PEMBAHASAN

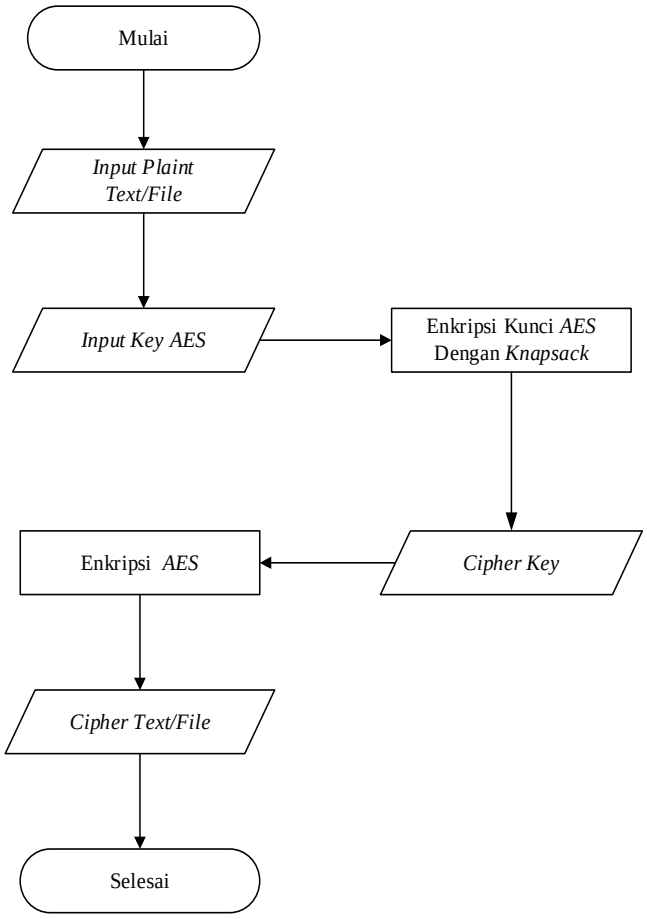
Masalah yang diselesaikan dalam tugas akhir ini antara lain adalah menggabungkan algoritma *Advanced Encryption Standard* (*AES*) dan algoritma *Knapsack,* yang nantinya digunakan untuk enkripsi dan dekripsi *text dan file*.

Tujuan pembuatan aplikasi ini adalah untuk mengamankan data (*text dan file*) sehingga data tersebut menjadi tidak dapat terbaca. Proses utama pada aplikasi ini adalah menggabungkan 2 algoritma *AES* dan *Knapsack* dengan kata lain bisa disebut

dengan algoritma hybrid. Pada waktu proses enkripsi kunci *AES* akan dienkripsi dengan algoritma *Knapsack* begitu pula pada saat mendekripsi. Adapun proses enkripsi dan dekripsi dalam aplikasi ini adalah sebagai berikut :

A. Enkripsi *Text* dan *File*

1. Pengguna memasukkan input berupa text atau file.
2. Masukkan kunci untuk mengenkripsi.
3. Lakukan enkripsi text atau file yang telah dimasukkan.
4. Diagram alir untuk enkripsi *text dan file* adalah sebagai berikut :

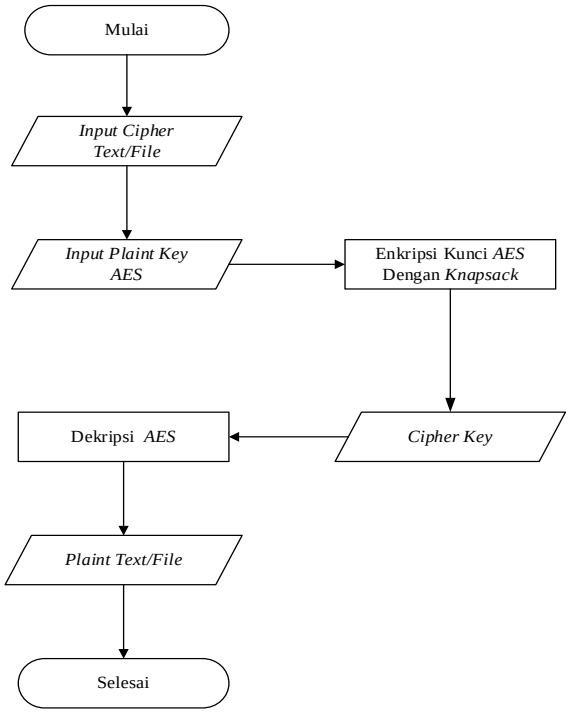


Gambar Enkripsi *Text dan file*

Seorang pengguna memasukkan data berupa text atau file kemudian pengguna juga memasukkan kunci yang digunakan untuk mengunci data, selanjutnya kunci tersebut akan dienkripsi dengan menggunakan algoritma *Knapsack*. Hasil dari enkripsi *Knapsack* ini dinamakan cipherkey dan cipherkey inilah yang akan di proses oleh algoritma *AES* pada saat enkripsi sampai mendapatkan hasil yang berupa *Ciphertext* maupun cipherfile.

B. Dekripsi *Text* dan *File*

1. Masukkan text atau file yang sudah terenkripsi.
2. Masukkan kunci yang digunakan saat enkripsi.
3. Lakukan dekripsi untuk text atau file yang telah dimasukkan.
4. Diagram alir untuk dekripsi *text dan file* adalah sebagai berikut :



Gambar Dekripsi *Text dan file*

Pada proses dekripsi seorang pengguna memasukkan databerupa *Ciphertext* atau cipherfile kemudian pengguna juga memasukkan kunci yang digunakan untuk mengunci data, selanjutnya kunci tersebut akan dienkripsi dengan menggunakan algoritma *Knapsack*. Hasil dari enkripsi *Knapsack* ini dinamakan cipherkey dan cipherkey inilah yang diproses algoritma *AES* pada saat dekripsi sampai mendapatkan hasil yang berupa *Plaintext* maupun plaintfile.

C. Penggabungan Algoritma AES dan Knapsack

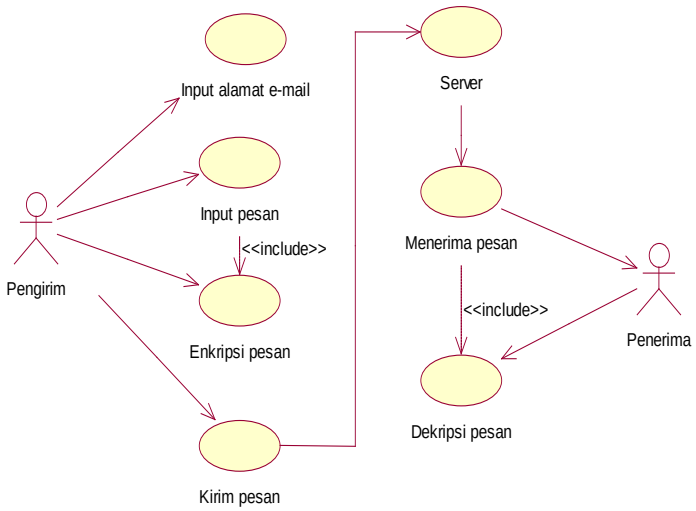
Langkah-langkah dalam menggabungkan dua algoritma tersebut adalah sebagai berikut:

- Kunci dari algoritma *AES* akan dienkripsi terlebih dahulu dengan algoritma *Knapsack.*
- Hasil enkripsi kunci dari algoritma *Knapsack* tersebut akan dimodulus dengan 256 bit.

- Hasil perhitungan modulo inilah yang akan menjadi kunci yang digunakan oleh algoritma *AES* pada saat enkripsi atau dekripsi.

D. Use Case Diagram

Use case diagram merupakan alat komunikasi tingkat tinggi untuk mewakili persyaratan sistem. Diagram menunjukkan interaksi antara pengguna dan pesan terenkripsi menggunakan kunci dari pengirim pesan didekripsi menggunakan kunci dari penerima pesan dengan sistem yang sedang dikembangkan. *Use case* digunakan untuk melihat antara sistem dengan pengguna atau disebut juga sebagai aktor. Berikut merupakan use case diagram dari enkripsi dan dekripsi pesan adalah sebagai berikut :



Gambar *Use Case Diagram* Enkripsi dan Dekripsi.

Pada gambar dapat dilihat seorang pengirim dapat melakukan penulisan pesan serta mengenkripsi pesan dan mengirim pesan, sedangkan penerima pesan dapat menerima pesan serta mendekripsi pesan dan juga menyimpan pesan yang telah diterima.

E. Penerapan Algoritma

Proses enkripsi adalah proses yang mengubah *Plaintext* menjadi *Ciphertext*. Sedangkan proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu mengubah *Ciphertext* menjadi *Plaintext*. Dalam melakukan proses dekripsi, isi file yang berupa *Ciphertext* harus diubah kembali menjadi pesan atau file asli. Berikut ini akan diberikan 1 contoh mengenai analisa terhadap file image.

Gambar Pinguin.jpeg

Gambar setelah dienkripsi (Pinguin.enc)

YMÀü?ÃÝHR§?>-\ç¬B*ŸB;3¸å˜;ý\â}¹¨ŒšS
ötÂÚµju‘öW®ªáTô1ñ½DÃÇpñ+«,nÌ¢ÖOÇ6
Û†Pø9Ï§$¨ÇH-ÃìùõŒ…HÚ(Ü¬Œó30%]†¨t€
Ã’ysÅý•"8ÿ/[¶!%□[I5<~!þ~›…¡•ôy!ú’…Š`Á
‰¸'Ñn=º¡†cWÙ×âÇu˜’T±$□à□Ø9òq©>Czo
Á÷e;‡ÇæxuVT“èsOõg¯$Z„‘5œ7Ñ“Òà/2,± u
‹ §¿ 5Öxüóþ¯•-*(Â€+è:éÝ Èœ"F,¡KZêAóô•¿
×ºâª ¤) °c Ö Ü.“n
□ÔyÿOlJÍ®,¿•ý_³«,À`‘éÔKu(†•+„òÓ7>&aÉ
üûRƒR)â~yÇI¤□ë&â¢?#RçšúþÊõÛ…‡@-h”

Gambar setelah didekripsi (Pinguin.jpeg)

Gambar Pinguin.jpeg

Berdasarkan contoh diatas penulis perlu melakukan analisa terhadap file tersebut dari segi waktu dan jenis file, terutama pada waktu proses enkripsi maupun dekripsi.

4. KESIMPULAN DAN SARAN

A. Kesimpulan

Dalam pembuatan aplikasi enkripsi dan dekripsi ini penulis dapat mengambil beberapa kesimpulan antara lain:

- Dengan cara menggabungkan algoritma *AES* dan *Knapsack* dimana algoritma *AES* digunakan untuk enkripsi dan dekripsi sedangkan algoritma *Knapsack* digunakan untuk enkripsi kunci dari algoritma *AES*, maka tingkat keamanan data atau informasi jauh lebih baik daripada menggunakan salah satu algoritma *AES* atau *Knapsack*.
- Implementasi gabungan algoritma *AES* dan *Knapsack* dilakukan dengan cara melakukan perhitungan modulo terhadap hasil enkripsi *Knapsack*, dan dilanjutkan enkripsi atau dekripsi dari algoritma *AES*.
- Ukuran file setelah dienkripsi akan berubah menjadi kecil dari ukuran sebelumnya. Sedangkan ukuran file akan kembali seperti semula setelah dilakukan proses dekripsi. Hal tersebut dikarenakan adanya proses kompresi dan dekompresi pada waktu proses enkripsi maupun dekripsi.

B. Saran

Dari beberapa kesimpulan yang didapatkan, maka saran yang dapat diberikan untuk pengembangan aplikasi enkripsi dan dekripsi *email* dengan algoritma *AES* dan *Knapsack* antara lain :

- Aplikasi dapat dikembangkan lagi untuk layanan berbasis web atau mobile.
- Dapat dikembangkan dalam aplikasi yang berupa add-on dalam web browser yang terintegrasi pada sebuah *email*.

REFERENSI

[1] Ariyus, Dony ,Pengantar Ilmu Kriptografi: Teori Analisis dan Implementasi, Andy, Amikom Yogyakarta.

[2] Munir, Rinaldi. 2006. Diktat Kuliah IF5054 *Advanced Encryption Standard* (*AES*). Program Studi Teknik Informatika, STEI, Institut Teknologi Bandung.

[3] Munir, Rinaldi. 2006. Diktat Kuliah IF5054 Algoritma *Knapsack*. Program Studi Teknik Informatika, STEI, Institut Teknologi Bandung.

[4] Prasetyo, Didik Dwi. 2006, Pemrograman Aplikasi Database Dengan Visual Basic .Net 2005 dan MS Access, PT Elex Media Komputindo, Jakarta.

[5] Stamp, Mark. 2011. Information Security: Principles and Practice, John Wiley & Sons, Canada.

[6] http://msdn.microsoft.com/library/vstudio