Pembangkit Stego-Teks Sederhana untuk Implementasi Steganografi

Halida Astatin (13507049)
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika,
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung, email: if17049@students.if.itb.ac.id

Abstrak *– Kriptografi merupakan ilmu yang mempelajari cara-cara menyembunyikan pesan. Terdapat berbagai teknik kriptografi, baik kriptografi klasik maupun modern. Dengan semakin maraknya penggunaan kriptografi, muncul semakin banyak kriptanalis yang dapat memecahkan cipherteks, sehingga berkembang sebuah metode penyembunyian pesan baru, yaitu steganografi. Steganografi adalah ilmu dan seni menyembunyikan informasi dengan cara menyisipkan suatu pesan rahasia di dalam pesan lain. Steganografi memungkinkan kita untuk menyembunyikan pesan dalam berbagai bentuk, seperti teks, gambar, audio, atau bahkan video. Bentuk steganografi yang paling sederhana adalah dalam bentuk teks, atau disebut juga dengan stego-teks. Konsep penyembunyian pesan dalam bentuk stego-teks relatif cukup sederhana, sehingga memungkinkan untuk membuat suatu pembangkit stego-teks otomatis yang dapat mengubah pesan asli menjadi stego-teks dengan memanfaatkan suatu kamus yang dirancang khusus untuk program tersebut.*

Kata Kunci*: kriptografi, steganografi, pembangkit stego-teks, kamus khusus*

1. PENDAHULUAN

Saat ini, teknologi informasi telah berkembang dengan pesat dan menjadi bagian dari kehidupan sehari-hari masyarakat. Cepatnya perkembangan teknologi ini tentu saja pada akhirnya menyebabkan peningkatan dalam pemanfaatan teknologi, salah satunya dalam hal pertukaran data melalui jaringan. Dalam hal ini, muncul sebuah isu yang cukup penting, yaitu isu keamanan data. Jaringan adalah sesuatu yang sangat rentan untuk diakses oleh siapapun, sehingga ketika terjadi pertukaran data dalam jaringan, keamanan data tersebut relatif rendah.

Selama ini, pemanfaatan algoritma-algoritma kriptografi, terutama algoritma modern yang bekerja dalam mode bit, merupakan salah satu metode yang diandalkan dalam usaha menjaga kerahasiaan data. Namun demikian, seiring makin berkembang dan maraknya krptografi, muncul semakin banyak kriptanalis yang dapat memecahkan pesan-pesan yang terenkripsi dengan metode mereka masing-masing. Banyaknya kriptanalis yang memanfaatkan metode masing-masing dalam usahanya memecahkan cipherteks yang dikirimkan menyebabkan tidak mungkin bagi kita untuk membuat suatu cipherteks yang terjamin keamanannya dari serangan semua kriptanalis.

Selain itu, kebutuhan untuk menyembunyikan pesan tanpa orang lain tahu ada pesan tersembunyi di dalamnya juga tidak dapat terakomodasi oleh kriptografi sederhana. Pesan terenkripsi dari hasil pengaplikasian algoritma kriptografi sederhana, baik algoritma klasik maupun modern, biasanya menghasilkan suatu cipherteks yang tidak dapat dimengerti sehingga orang yang melihatnya akan menyadari bahwa terdapat suatu pesan tersembunyi di dalamnya. Untuk mengatasi permasalahan ini, digunakan steganografi. Dengan menggunakan steganografi, pesan yang dienkripsi masih dapat terbaca sebagai teks normal sehingga tidak menimbulkan kecurigaan. Salah satu bentuk steganografi adalah stego-teks. Stego-teks memiliki pola-pola tertentu sehingga mungkin dibuat sebuah pembangkit stego-teks. Dalam makalah ini akan dibahas mengenai suatu pembangkit stego-teks sederhana.

2. DEFINISI

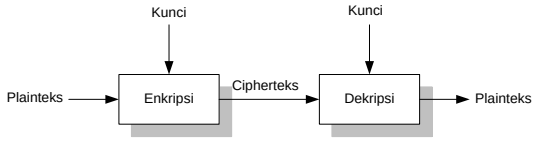
Sebelum berangkat lebih jaud membahas mengenai pembangkit stego-teks sederhana, perlu dijabarkan lebih lanjut definisi dari kriptografi, steganografi, dan stego-teks itu sendiri yang akan digunakan untuk menyelesaikan permasalahan.

2.1. Kriptografi

Kriptografi berasal dari bahasa Yunani κρυπτός (kryptos) yang berarti tersembunyi atau rahasia, dan γράφω (gráphō) yang berarti menulis. Kriptografi adalah suatu ilmu yang digunakan untuk menyamarkan atau menyembunyikan pesan. Kriptografi digunakan umumnya untuk menjaga keamanan pesan. Kemanan pesan yang dimaksud dalam hal ini harus mencakup beberapa aspek, yaitu

kerahasiaan, integritas, autentikasi, dan non-repudiasi. Aspek kerahasiaan berarti dapat menjaga pesan dari pihak yang tidak memiliki otoritas untuk melihat isi pesan tersebut, sedangkan integritas berarti menjaga agar isi pesan tidak berubah secara tidak sah. Autentikasi berkaitan dengan masalah identifikasi pihak yang mengakses data, sedangkan non-repudiasi berkaitan dengan pihak yang mengirimkan data. Dengan adanya non-repudiasi, pihak yang mengirimkan data tidak dapat menyangkal bahwa memang dialah yang mengirimkan data tersebut.

Kriptografi mempelajari berbagai teknik mengubah teks awal (plainteks) menjadi teks yang terenkripsi (cipherteks). Proses penerjemahan plainteks ke cipherteks dan sebaliknya dilakukan dengan menggunakan suatu kunci. Kunci yang digunakan mungkin sama atau berbeda. Berdasarkan sama atau tidaknya kunci yang digunakan untuk enkripsi dan dekripsi, algoritma kriptografi dibedakan menjadi dua, yaitu kriptografi kunci-simetris dan kriptografi kunci-asimetris. Algoritma kriptografi yang menggunakan kunci yang sama untuk enkripsi maupun dekripsi disebut algoritma kriptografi kunci-simetris, contohnya algoritma DES dan blowfish, sedangkan algoritma kriptografi yang menggunakan kunci yang berbeda untuk enkripsi dan dekripsinya disebut algoritma kriptografi kunci-asimetris, contohnya algoritma RSA.



Gambar 1 Skema Umum Kriptografi

Berdasarkan zaman penggunaannya, algoritma kriptografi dapat dibedakan menjadi kriptografi klasik dan kriptografi modern. Kriptografi klasik adalah kriptografi yang menggunakan algoritma berbasis karakter. Teknik ini sudah digunakan sejak jaman dahulu kala, sehingga algoritma ini sebenarnya tidak membutuhkan penggunaan komputer dalam implementasinya, cukup menggunakan pena dan kertas saja. Algoritma kriptografi klasik termasuk dalam kriptografi kunci-simetri. Terdapat dua jenis algoritma kriptografi klasik, yaitu cipher substitusi dan cipher transposisi. Inti dari cipher substitusi adalah mengganti suatu huruf dengan huruf lainnya dengan cara menggeser tiap huruf alfabet, contohnya Caesar Cipher. Di sisi lain, cipher transposisi bekerja dengan prinsip menukar posisi huruf-huruf yang ada dalam plainteks (seperti permainan anagram). Kedua algoritma ini dapat pula digabungkan menjadi Super-enkripsi.

Namun demikian, enkripsi menggunakan algoritma kriptografi klasik termasuk algoritma yang mudah untuk ‘diserang’. Jika seseorang berusaha untuk memecahkan makna suatu cipherteks, yang dibutuhkan hanya kunci atau contoh pasangan plainteks–cipherteks. Untuk itu, algoritma kriptografi kemudian dibuat menjadi semakin kompleks, contohnya pada algoritma kriptografi modern.

Algoritma kriptografi modern bekerja dalam mode bit. Kunci, plainteks, dan cipherteks diproses sebagai rangkaian bit. Operasi yang paling banyak digunakan adalah operasi XOR. Algoritma ini tetap menggunakan gagasan yang digunakan pada algoritma kriptografi klasik, yaitu memanfaatkan prinsip substitusi dan transposisi. Perkembangan algoritma ini didorong oleh penggunaan komputer digital untuk keamanan pesan, karenanya algoritma ini beroperasi dalam mode biner. Terdapat dua jenis algoritma berbasis bit, yaitu stream cipher dan block cipher. Stream cipher beroperasi pada bit tunggal, dan melakukan enkripsi maupun dekripsi secara bit per bit. Block cipher beroperasi pada blok bit, dan enkripsi maupun dekripsi dilakukan secara blok per blok.

Munculnya kriptografi sebagai suatu ilmu untuk menyembunyikan atau menyamarkan isi pesan mendorong munculnya Kriptanalisis, yaitu ilmu untuk menemukan isi pesan yang tersembunyi. Kriptanalisis dapat digunakan untuk mengambil isi cipherteks tanpa otoritas, namun dapat pula digunakan untuk menguji kekuatan suatu algoritma kriptografi. Semakin banyaknya kriptanalis yang handal dalam menemukan isi pesan memunculkan kebutuhan akan sautu teknik penyembunyian pesan yang lebih baik dan lebih kreatif lagi, agar keamanan pesan tetap terjamin. Dari kebutuhan inilah muncul steganografi sebagai salah satu metode penyembunyian pesan.

2.2. Steganografi

Steganografi berasal dari bahasa Yunani steganos yang berarti tersamarkan, dan γράφω (gráphō) yang berarti menulis. Steganografi disebut sebagai ilmu atau seni menulis pesan secara terselubung (*concealed writing*). Tercatat bahwa istilah ini pertama kali digunakan pada tahun 1499 oleh Johannes Trithemius dalam *Steganographia*, suatu *treatise* tentang kriptografi dan steganografi yang disamarkan sebagai buku tentang sihir. Pada umumnya pesan akan tampil sebagai suatu gambar, artikel, atau *covertext* lainnya.

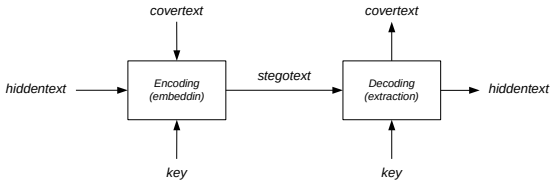
Steganografi ada sejak 440 SM, ketika Demaratus akan mengirimkan peringatan tentang serangan kepada Yunani dengan cara menuliskan hal tersebut di atas panel kayu dan menutupinya dengan

minyak-minyakan. Contoh lainnya adalah ketika Histiaeus mencukur habis rambut pelayannya yang paling setia kemudian mentato pesan rahasia disana. Setelah rambut pelayan itu tumbuh, pesan itu akan tersembunyi.

Keunggulan steganografi dibandingkan dengan kriptografi adalah pesan-pesan yang disembunyikan menggunakan steganografi tidak akan menarik perhatian. Teks yang dienkripsi dengan kriptografi, apapun algoritma yang diaplikasikan, akan terlihat tidak terbaca dan tidak dapat dimengerti, sehingga pihak yang melihatnya akan dapat langsung mencurigai bahwa pesan tersebut mengandung pesan lain yang tersembunyi di dalamnya. Karena itu, jika kriptografi dapat melindungi isi dari suatu pesan, steganografi dapat dikatakan mampu memproteksi pesan maupun kedua belah pihak yang bertukar pesan tersebut.

Steganografi mencakup penyembunyian informasi di dalam file komputer. Dalam steganografi digital, komunikasi elektronik melibatkan manipulasi kode pada layer transport, misalnya file dokumen, file gambar, file program, atau bahkan protokol. Media digital ideal untuk transmisi steganografi karena ukurannya yang besar. Sebagai contoh sederhana, seorang pengirim pesan dapat mengubah suatu file gambar dan menyesuaikan setiap pixel ke-100 untuk berkorespondensi dengan sebuah huruf pada alfabet. Perubahan yang akan terjadi pada gambar akan sangat halus sehingga orang yang tidak tahu bahwa gambar tersebut mengandung pesan rahasia tidak akan memperhatikan perubahan tersebut.

Steganografi memiliki beberapa properti dalam menyembunyikan pesan. Properti-properti tersebut adalah *embedded message* (*hiddentext*), *cover-object* (*covertext*), *stego-object* (*stegotext*), dan *stego-key*. *Embedded message* (*hiddentext*) adalah pesan yang disembunyikan. Pesan ini dapat berupa teks biasa maupun media seperti gambar, audio, video, dan lain-lain. *Cover-object* (*covertext*) adalah pesan yang digunakan untuk menyembunyikan *embedded message*. Sama halnya dengan embedded message, cover-object dapat berupa teks, gambar, audio, video, maupun bentuk file lainnya. *Stego-object* adalah pesan yang sudah berisi embedded message, sedangkan *stego-key* adalah kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari *stegotext*.



Gambar 2 Skema Umum Steganografi

2.3. Stego-teks

Salah satu bentuk steganografi yang paling sederhana adalah menyembunyikan teks dalam teks. Pemodelan steganografi dalam bentuk seperti ini ada dalam *Prisoners' Problem* yang terdapat pada *"The Prisoner's Problem and the Subliminal Channel"* oleh Gustavus Simmons. Dalam *Prisoners' Problem*, diceritakan bahwa Alice dan Bob adalah dua orang tahanan yang dikurung dalam dua sel yang terpisah jarak. Mereka diperbolehkan untuk berkomunikasi dengan menuliskan pesan pada kertas. Mereka berdua akan menyusun rencana untuk kabur dari penjara, namun seluruh bentuk komunikasi yang mereka lakukan akan dijaga oleh seorang sipir penjara bernama Eve. Jika Eve mendeteksi adanya konspirasi atau rencana melarikan diri sekecil apapun pada pesan, maka Alice dan Bob akan dikirim ke sel isolasi. Dalam hal ini, dibutuhkan suatu cara dimana Alice dan Bob dapat mengirimkan pesan rahasia tanpa Eve sadar bahwa pesan tersebut mengandung sebuah pesan rahasia.

Pada kasus ini, misalkan Alice akan mengirim pesan pada Bob yang berisi 'lari jam satu'. Jika Alice mengenkripsi pesan tersebut menggunakan suatu algoritma kriptografi, cipherteks yang dihasilkan akan berbentuk barisan huruf yang tidak dapat dimengerti dan pasti memancing kecurigaan Eve. Untuk itulah digunakan steganografi, dalam bentuk stego-teks. Jika pesan yang dikirim Alice disembunyikan dalam suatu kalimat yang menyembunyikan pesan 'lari jam satu' dalam huruf-huruf pertamanya, misalnya dalam kalimat '**l**upakan **a**sal **r**umor **i**tu, **j**aga **a**gar **m**atamu **s**ehat **a**tau **t**urunkan **u**banmu', maka Eve tidak akan curiga.

Pada contoh yang diberikan di atas, properti-properti steganografi pesan tersebut adalah sebagai berikut:

- √ *Hiddentext:* Lupakan asal rumor itu, jaga aga matamu sehat atau turunkan ubanmu
- √ *Covertext*: upakan sal umor tu aga aga atamu ehat tau turunkan banmu
- √ *Hiddentext*: Lari jam satu
- √ *Stegotext*: Lupakan asal rumor itu, jaga agar matamu sehat atau turunkan ubanmu

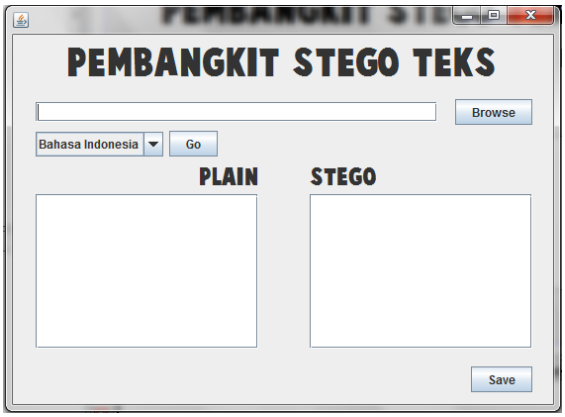
Selain dalam kasus yang telah dijelaskan, masih banyak kasus lain yang dapat memanfaatkan stego-teks ini. Metode yang digunakanpun dapat divariasikan dengan berbagai algoritma. Adanya pola dalam pembuatan stego-teks ini juga memungkinkan dibuatnya suatu pembangkit stego-teks sederhana yang dapat secara otomatis membentuk stego-teks dari suatu plainteks yang dimasukkan pengguna.

3. PEMBANGKIT STEGO-TEKS

Saat ini, kebutuhan akan stego-teks sebagai media untuk menyamarkan pesan rahasia dapat dikatakan sudah semakin tinggi. Alasan pertama adalah karena semakin banyak kriptanalis yang mampu memecahkan berbagai algoritma yang paling rumit sekalipun. Alasan kedua adalah adanya kebutuhan untuk membuat suatu pesan rahasia tanpa pihak yang melihat sadar bahwa didalamnya tersimpan pesan rahasia. Disinilah peran stego-teks dapat menggantikan cipherteks hasil enkripsi dengan algoritma kriptografi biasa.

Dengan meningkatnya kebutuhan menggunakan stego-teks, adanya alat yang dapat memudahkan dalam pembuatan stego-teks itu sendiri akan sangat membantu. Stego-teks memiliki pola-pola tertentu dalam pembentukannya, sehingga memungkinkan seorang *programmer* untuk membuat suatu pembangkit stego-teks sederhana dengan spesifikasi dan batasan tertentu.

3.1. Spesifikasi Program

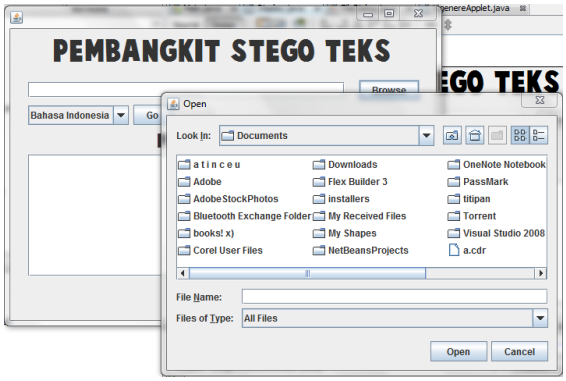


Gambar 3 Layar Utama Program

Program ini dibuat dengan menggunakan bahasa Java dan memanfaatkan IDE NetBeans 6.8. Lingkungan sistem operasi yang digunakan adalah Windows 7.

Idenya adalah membuat suatu pembangkit stego-teks yang menerima masukan sebuah plainteks, kemudian mengkonversi plainteks tersebut menjadi sebuah stego-teks. Program ini beroperasi dalam mode karakter, sehingga file yang dapat dimanipulasi oleh program ini hanyalah file teks. Program dapat membaca suatu file eksternal kemudian menampilkan isinya pada suatu *textbox* pada layar program. Selain dari file eksternal, program dapat juga membaca masukan dari *textbox* yang tersedia.

Pada layar utama program, tersedia sebuah field teks dimana pengguna dapat memasukkan alamat file, dan di sampingnya juga terdapat tombol untuk memunculkan open file dialog. Pengguna kemudian dapat memilih bahasa yang akan digunakan pada stego-teks. Pilihan bahasa ini menentukan kamus mana yang akan di-*load* oleh program untuk mencari *covertext* untuk digunakan pada stego-teks. Di bawahnya terdapat dua buah *textarea*, satu untuk menampilkan plainteks dan satunya lagi untuk menampilkan stego-teks. Kemudian di bagian layar paling bawah terdapat sebuah tombol untuk menyimpan file.



Gambar 4 Open File Dialog pada Program

Aturan pengubahan yang digunakan untuk mengubah *hiddentext* menjadi stego-teks adalah dengan menyamarkan huruf-huruf *hiddentext* di dalam stego-teks sesuai posisinya pada plainteks. Sumber *covertext* yang digunakan adalah suatu kamus khusus yang telah dikelompokkan dan ditandai sesuai tipe komponen kalimatnya, seperti subjek, objek, dan kata kerja atau predikat. Program akan membuka kamus ketika pengguna memilih bahasa, kemudian mengiterasi isi kamus sesuai dengan masukan pengguna. Hasil stego-teks yang dihasilkan program akan berupa sekumpulan kalimat yang masing-masing terdiri dari tiga kata dan diakhiri tanda titik.

Contoh pengubahan dari plainteks menjadi stego-teks dengan menggunakan aturan ini adalah sebagai berikut: untuk plainteks “lapor”, pembangkit stego-teks akan menghasilkan “**l**ampu j**a**tuh. Se**p**rai mem**o**tong kasu**r**.” Tentu saja kalimat yang terbentuk tidak selamanya logis, namun stego-teks yang dibangkitkan sudah dapat digunakan. Untuk plainteks yang lebih panjang, akan dibutuhkan penyesuaian-penyesuaian lebih lanjut.

Mempertimbangkan panjangnya plainteks yang mungkin dimasukkan pengguna dan panjang kata secara umum dalam bahasa Indonesia serta bahasa Inggris, maka dibutuhkan aturan khusus tentang penentuan kata yang dicari. Seperti yang sudah disebutkan sebelumnya, huruf-huruf pada *hiddentext* akan disembunyikan sesuai posisinya pada plainteks. Huruf pertama kalimat plainteks

akan disembunyikan pada huruf pertama kata pertama, huruf ke-dua kalimat plainteks akan disembunyikan pada huruf ke-dua kata ke-dua, dan seterusnya. Namun, masalah timbul jika panjang plainteks mencapai angka puluhan atau bahkan ratusan huruf. Pada umumnya kata dalam bahasa Inggris dan Indonesia hanya memiliki paling banyak sepuluh huruf. Untuk itu, nilai posisi yang dicari akan berulang setiap kelipatan lima huruf. Artinya, setelah menyembunyikan huruf ke-lima plainteks pada huruf ke-lima kata ke-lima, program akan menyembunyikan huruf ke-enam plainteks pada huruf pertama kata ke-enam.

3.2. Batasan Program

Pengembangan program ini masih terbatas oleh beberapa kendala, sehingga memiliki berbagai keterbatasan. Hal utama yang membatasi dalam pengembangan program ini adalah kamus khusus yang dibutuhkan oleh program. Dalam program ini keberadaan kamus sangat penting karena merupakan sumber *covertext* yang akan digunakan dalam membuat stego-teks. Sayangnya, proses pembuatan kamus khusus untuk program ini cukup menyita waktu karena kamus yang digunakan pada program harus dikelompokkan dan ditandai berdasarkan tipe komponen kalimatnya.

Selain itu, struktur kalimat setiap bahasa berbeda-beda, sehingga pengelompokkan kata dalam kamus pun harus disesuaikan dengan bahasa yang digunakan. Dalam mengelompokkan kata-kata ini dapat dilakukan dengan meletakkan suatu karakter khusus di awal kata (misalnya tanda @ untuk subjek, # untuk objek, dan * untuk predikat atau kata kerja), dan dapat pula dilakukan dengan memisahkan kelompok kata berdasarkan posisinya dalam file kamus.

Selain itu, karena program ini beroperasi dalam mode karakter, algoritma yang digunakan hanya dapat diimplementasikan pada file teks. Tipe file lain seperti file gambar, audio, atau video harus diproses dalam mode bit sehingga tidak dapat ditangani oleh program ini.

Keterbatasan lain muncul ketika program tidak dapat menemukan kata yang dapat menjadi *covertext* untuk huruf yang sedang diproses. Jika menemukan kondisi seperti ini, program akan menuliskan huruf apa adanya dalam stego-teks.

Dalam makalah ini, yang akan dibahas selanjutnya hanya metode langkah-langah algoritmik untuk pengubahan dari plainteks menjadi stego-teks. Langkah-langkah tersebut mencakup pemecahan plainteks berdasarkan posisi huruf, pencarian kata yang dapat menjadi covertext untuk huruf tersebut, membangkitkan stego-teks kata per kata, kemudian mengembalikan stego-teks secara keseluruhan kepada pengguna.

3.3. Algoritma Utama Program

Sebelum beranjak kepada pembuatan algoritma program, sebelumnya perlu ditentukan format yang digunakan dalam kamus yang akan diiterasi isinya. Dalam makalah ini, kamus yang digunakan ditandai dengan tanda @ untuk subjek, # untuk objek, dan * untuk predikat atau kata kerja. Selain itu, kata-kata yang telah ditandai akan dikelompokkan dalam file kamus, dan antar tipe komponen kalimat dibatasi dengan tanda =.

@ ayah	@ father
@ ibu	@ mother
=	=
* bernyanyi	* match
* masak	* is
=	=
# lagu	# pattern
# buku	# tired

Gambar 5 Contoh Isi Kamus dalam Bahasa Indonesia dan Bahasa Inggris

Algoritma yang diaplikasikan akan mengubah plainteks menjadi suatu stego-teks dengan cara menyamarkan huruf-huruf pada plainteks sesuai dengan posisinya pada plainteks. Setiap huruf pada plainteks akan diwakili oleh sebuah kata dalam stego-teks, dan posisi huruf yang disamarkan pada kata di stego-teks akan ditentukan oleh posisi huruf pada plainteks. Jika panjang plainteks lebih dari 5 karakter, maka nilai posisi yang menentukan posisi huruf pada plainteks dimodulasi dengan angka 5. Implementasi algoritma tersebut adalah sebagai berikut:

```
procedure Stego (input String Plain)

Deklarasi:
procedure BacaKamus (int Bahasa)
{Membaca file eksternal kamus dan
menyimpannya ke dalam struktur data
internal sementara. Nilai integer
yang dimasukkan menentukan file apa
yang di-load}

String[] CariKata(int pos, char x,
int tipe)
{Mencari kata-kata dalam kamus yang
memiliki karakter x pada posisi pos
dengan tipe yang sesuai. Tipe 1
berarti subjek, 2 berarti
predikat/kata kerja, dan 3 berarti
objek. Fungsi ini mengembalikan array
of string yang berisi kata-kata
tersebut}
```

```
Boolean IsEmpty (String[] Arr)
{Menerima sebuah array of string dan
mengembalikan true jika array
tersebut kosong dan sebaliknya}

String PilihRandom(String[] Pilihan)
{Memilih satu string secara acak dari
suatu array of String}

Algoritma:

Stego: String
Stego ← ""
Bahasa ← <bahasa pilihan pengguna>

BacaKamus(bahasa)
for i = 1 to plainteks.length do
   a ← i mod 5
   Kata[] ← CariKata(a, plainteks[i],
     (i mod 3))
   if (IsEmpty(Kata[])
     Stego ← Stego + plainteks[i]
   else
     KataStego ← PilihRandom (Kata[])
     Stego ← Stego + KataStego
   if ((i mod 3) == 0)
     Stego ← Stego + ". "
   else
     Stego ← Stego + " "

{kondisi berhenti: seluruh huruf pada
plainteks telah teriterasi, Stego
sudah berisi stego-teks yang dapat
digunakan}
```

Dalam algoritma yang telah disebutkan, pertama-tama perlu didefinisikan sebuah prosedur dan tiga buah fungsi. Prosedur yang dideklarasikan akan membaca kamus berdasarkan bahasa yang dipilih dan menyimpannya ke dalam suatu struktur data internal berupa tiga buah *array of string*, masing-masing satu untuk tipe komponen kalimat subjek, predikat, dan objek. Setelah proses iterasi berakhir, ketiga *array of string* ini akan dikosongkan lagi agar tidak boros memori.

Fungsi yang dibutuhkan dalam algoritma ini, pertama adalah fungsi CariKata. Fungsi CariKata menerima masukan integer yang melambangkan posisi huruf yang dicari, karakter yang dicari pada posisi tersebut, serta tipe kata yang dicari. Tipe kata akan menentukan *array* mana yang akan diiterasi isinya untuk mendapatkan kata untuk stego-teks. Fungsi ini mengembalikan suatu *array of string* yang berisi kata-kata yang memenuhi syarat yang disebutkan pada parameter fungsi.

Fungsi lain yang dibutuhkan adalah fungsi untuk mengecek apakah *array* yang dihasilkan dari fungsi CariKata menghasilkan kata yang valid atau tidak. Fungsi yang digunakan adalah fungsi IsEmpty yang menerima parameter *array of string* dan mengembalikan *true* jika *array* kosong dan mengembalikan *false* jika *array* ada isinya.

Selanjutnya, jika iterasi yang dilakukan hanya iterasi biasa dan menggunaan kata pertama yang ditemukan, maka akan terjadi banyak redundansi kata dalam stego-teks yang dihasilkan. Untuk itu, program mencari seluruh kata yang memenuhi syarat yang disebutkan pada parameter fungsi CariKata, menghasilkan *array of string* kata-kata yang valid, kemudian *array* itu akan digunakan sebagai parameter fungsi PilihRandom. Fungsi PilihRandom berfungsi untuk memilih secara acak satu kata dari seluruh elemen *array of string* yang menjadi parameternya. Dengan memanfaatkan fungsi ini, diharapkan kata-kata yang terpilih untuk digunakan pada stego-teks akan lebih beragam dan dapat dihindari redundansi berlebihan.

Pada program utama, cukup dilakukan pemanggilan fungsi-fungsi yang telah dibuat. Disiapkan sebuah *string* kosong untuk menampung stego-teks yang dihasilkan algoritma. Selanjutnya, langkah pertama yang dilakukan adalah membaca isi kamus sesuai bahasa yang dipilih pengguna. Program kemudian mengiterasi karakter yang terdapat pada plainteks satu per satu. Untuk setiap karakter, dicari kata-kata yang memenuhi fungsi CariKata pada posisi huruf tersebut dimodulasi dengan nilai 5. Jika fungsi CariKata menghasilkan *array* kosong, maka karakter yang sedang diperiksa langsung ditambahkan pada stego-teks, sedangkan jika fungsi CariKata menghasilkan *array* yang tidak kosong, dipanggil fungsi PilihRandom untuk menentukan kata mana yang ditambahkan pada stego-teks. Setiap penambahan tiga kata pada stego-teks, ditambahkan sebuah karakter "." (titik) pada stego-teks tersebut. Hasil akhir stego-teks yang dihasilkan akan berupa sekumpulan kalimat yang masing-masing terdiri atas tiga kata.

4. KESIMPULAN

Kriptografi adalah suatu ilmu atau seni untuk menyembunyikan suatu pesan rahasia. Terdapat berbagai macam algoritma kriptografi; berdasarkan kunci yang digunakan, terbagi menjadi algoritma kriptografi kunci-simetrik dan algoritma kriptografi kunci-asimetrik, sedangkan berdasarkan zamannya dibagi menjadi algoritma kriptografi klasik dan algoritma kriptografi modern. Saat ini, seiring dengan berkembangnya kriptografi, semakin banyak kriptanalis handal yang dapat menemukan algoritma pemecahan untuk mendekripsi pesan

yang telah dienkripsi menggunakan suatu algoritma kriptografi. Untuk itu, dibutuhkan suatu cara agar pihak luar tidak menyadari keberadaan pesan rahasia dalam pesan yang kita kirimkan. Hal ini dapat diakomodasi dengan menggunakan steganografi, yaitu ilmu atau seni menulis pesan secara terselubung.

Keuntungan penggunaan steganografi dibandingkan dengan kriptografi sederhana adalah bahwa steganografi tidak hanya melindungi pesan yang disembunyikannya, tapi juga kedua pihak yang bertukar pesan rahasia. Ketika seseorang menyembunyikan pesan dengan steganografi, pihak lain yang melihat hasil enkripsi pesan tersebut tidak akan mencurigai adanya pesan rahasia yang tersimpan. Salah satu bentuk steganografi yang paling sederhana adalah penyembunyian teks dalam teks. Penggunaan steganografi dengan cara ini memiliki pola-pola tertentu sehingga tidak menutup kemungkinan bagi seorang *programmer* untuk membuat sebuah pembangkit stego-teks sederhana untuk membantu pembentukan stego-teks.

Pembangkitan stego-teks dengan program yang dibuat bekerja dengan prinsip mengubah satu karakter pada plainteks menjadi satu kata baru pada stego-teks, tergantung pada karakter itu sendiri dan posisinya pada plainteks. Namun demikian, program ini masih sangat terbatas kendala bahasa, karena dibutuhkan kamus khusus yang memisahkan kata-kata dalam bahasa tersebut dalam kelompok-kelompok subjek, predikat, dan objek. Perbedaan struktur kalimat tiap-tiap bahasa juga memperumit dalam pembuatan program.

DAFTAR REFERENSI

[1] Amira, Hapsari Muthi. 2009. *Makalah Struktur Diskrit: Studi Steganografi pada Image File.* Institut Teknologi Bandung.

[2] Munir, Rinaldi. 2004. *Bahan Kuliah IF3058 Kriptografi*. Departemen Teknik Informatika, Institut Teknologi Bandung.

[3] Setiawan, Rachmansyah Budi. 2009. *Makalah Struktur Diskrit: Penggunaan Kriptografi dan Steganografi Berdasarkan Karakteristik Keduanya*. Institut Teknologi Bandung.

[4] Künnemann, Robert. 2007. *Planning a Jailbreak: Use Steganography*. http://docs.google.com diakses 23 Maret 2010 pukul 23:41.

[5] http://en.wikipedia.org/wiki/steganography diakses 20 Maret 2010 pukul 22:19.

[6] http://en.wikipedia.org/wiki/cryptography diakses 20 Maret 2010 pukul 22:22.