SOLUSI ALTERNATIF PENGAMANAN INFORMASI PENTING/RAHASIA DENGAN MENYEMBUNYIKAN TEKS BERITA DI DALAM SPAM

E. Yon Handri
Bidang Perangkat Lunak Deputi III, Lembaga Sandi Negara
E-MAIL :nguyen_kai@yahoo.com

Abstrak
Pada umumnya informasi penting atau rahasia diamankan dengan menggunakan password atau teknik enkripsi tertentu agar tidak dapat diketahui oleh pihak yang tidak berhak, baik secara online dari internet maupun offline dari komputer lokal. Keberadaan informasi penting yang dienkripsi menjadi target utama dalam proses pencarian data oleh pihak yang tidak berkepentingan di dunia internet, sebab diyakini data yang terenkripsi tersebut tentunya merupakan informasi penting atau rahasia. Ketika beberapa pihak berfokus pada pengamanan informasi dengan enkripsi, terdapat solusi alternatif untuk mengamankan informasi ke dalam bentuk spam atau bentuk berita lain. Spam atau email sampah dianggap mengganggu pengguna internet dan sering diacuhkan dalam trafik internet sehingga menjadi peluang dalam mengamankan informasi di dalamnya. Paper ini menjelaskan bagaimana suatu informasi dalam bentuk teks dapat disembunyikan ke dalam spam berbahasa Indonesia, penjelasanan mimic function sebagai inti dari proses, dan tingkat keamanan mimic function tersebut, serta future works dalam perancangan aplikasi ke depan berbasis java mobile. Solusi alternatif pengamanan informasi ini diharapkan mampu mendukung jalannya e-government dengan segala tantangan yang ada.
Keyword : *spam, enkripsi, mimic function, steganografi ,pengamanan informasi*

1. PENDAHULUAN

Setiap informasi yang ingin dijaga kerahasiaanya memerlukan pengamanan agar isi dari informasi tersebur tidak jatuh kepada pihak yang tidak berhak memilikinya. Umunya pengamanan informasi lebih banyak menggunakan password atau teknik enkripsi tertentu dalam ilmu kriptografi. Seiring berkembangnya pengamanan informasi, khususnya dengan menggunakan enkripsi, berkembang pula kriptanalisis yaitu cara membongkar informasi hasil enkripsi. Adanya kriptanalisis ini menyebabkan informasi yang terenkripsi menjadi target dalam lalu lintas data baik melalui internet maupun jaringan komputer yang lain untuk diketahui isinya. Apalagi bentuk informasi yang terenkripsi dalam bentuk file data mudah dikenali karena file tidak terbaca dan isi data acak. Oleh karena itu, ketika beberapa pihak berfokus pada pengamanan informasi dengan enkripsi, terdapat solusi alternatif pengamanan informasi menggunakan spam dengan konsep informasi rahasia yang diamankan menjadi bentuk informasi lain bisa dibaca atau tidak acak.

Salah satu alasan penggunaan spam dalam pengamanan informasi adalah anggapan bahwa spam merupakan email sampah yang mengganggu kenyamanan membuka email. Spam biasanya berisi penawaran-penawaran produk yang kurang penting dan tidak diketahui siapa pengirimnya. Bahkan beberapa spam mengandung unsur penipuan. Indonesia sendiri merupakan salah satu negara yang menjadi target serbuan spam. Berdasarkan data dari International Data Center (IDC), rata-rata sekitar 70-80 persen inbox pengguna adalah spam. Akibatnya para pengguna internet cenderung sering mengacuhkan spam di dalam inbox-nya dan segera menghapusnya sesegera mungkin. Kecenderungan inilah yang dijadikan suatu kelebihan untuk menyembunyikan informasi rahasia ke dalam spam yaitu :

- Spam cenderung diacuhkan sehingga informasi rahasia di dalamnya dapat lepas dari target pencarian data-data terenkripsi melalui internet.
- Spam berisi kalimat-kalimat yang dapat dibaca (tidak acak) sehingga menghilangkan kesan bahwa spam hasil dari proses pengamanan informasi (enkripsi).

Proses transformasi dari pesan rahasia menjadi bentuk spam menggunakan *mimic function. Mimic function* merupakan salah satu teknik steganografi yang mentransformasikan pesan menjadi format lain namun pesan tersebut masih bisa dibaca secara gramatikal (tata bahasa). Hal ini berbeda dengan proses enkripsi yang merubah pesan menjadi bentuk yang acak.

2. TINJAUAN PUSTAKA

2.1 Definisi *Mimic function*

Menurut Peter Wayner dalam bukunya " *Dissapearing Cryptography Information Hiding : Steganography and Watermaking*", *mimic function* didesain untuk menyembunyikan informasi dengan merubahnya menjadi format lain yang tidak mencurigakan (innocent format). Sedangkan menurut wikipedia, *mimic function* merubah suatu file A sehingga diasumsikan secara statistikal merupakan file B, yaitu jika p(t,A) adalah probabilitas dari beberapa substring t pada A, maka *mimic function* f, menyimpan A sehingga p(t,f(A)) mendekati p(t,B) untuk semua string t dengan panjang kurang dari beberapa n.

Terdapat 3 (tiga) tipe yang berbeda dari *mimic function* untuk menghasilkan hierarki standart kompleksitas bahasa yaitu :

a. *Reguler Languages* (bahasa biasa)
   Pada tipe ini, *mimic function* merubah data menjadi string yang diambil dari himpunan bahasa biasa yang sudah ditentukan. Contoh tipe pertama ini adalah fungsi kompresi karena jika sebuah algoritma kompresi seperti Huffman mengkonversi sebuah fle dengan distribusi statistik menjadi bagian kecil atau file yang terdistribusi maka invers-nya dapat menkonversi file yang terdistribusi tersebut menjadi suatu file mimic secara statistik. Struktur internal pada tipe pertama ini memiliki kemiripan dengan homophonic cipher yang dikembangkan oleh Massey dan kawan-kawan.

b. *Context-free languages* atau *Context-free grammar (CFG)*
   Tipe kedua dari *mimic function* adalah merubah data menjadi string yang diambil dari bahasa dengan konteks bebas dan tidak ambigu. Bit-bit dari file data digunakan untuk memilih produksi *grammar* sehingga menghasilkan suatu string. Bit-bit tersebut dapat di-parsing kembali dari string dengan cara menemukan daftar produksi yang dibangkitkan untuk menghasilkan string tersebut. Jika hasil string tidak ambigu maka informasi rahasia dapat disembunyikan dan di-parsing kembali.

c. *Recursively-enumerable languages*
   Tipe ketiga dari *mimic function* ini menggunakan *turing machine* seperti yang digunakan dalam mesin sandi. Namun *mimic function* disini menggunakan bentuk *grammar* Van Wijngaarden atau disingkat VW *grammar*. VW *Grammar* dapat membentuk *context-free grammar* yang bersifat double dimana sebuah meta-level CFG memilih produksi yang diikuti oleh lower-level CFG.

2.2 Algoritma *Mimic function*

Penjelasan proses pengolahan informai/data pada *mimic function* dalam menyembunyikan informasi lebih mudah dijelaskan dengan tipe kedua *mimic function*. Tipe kedua ini dapat menciptakan suatu string yang secara tata bahasa (gramatikal) benar dan terlihat seperti bahasa manusia dalam hal ini berbentuk spam. Algoritma ini menggunakan *context-free grammar* (CFG) yang berarti bebas menentukan tata bahasa yang digunakan baik itu bahasa Inggris, Indonesia atau bahasa lain.

CFG terdiri dari 3 (tiga) bagian yang berbeda yaitu :

a. Terminal
   Terminal adalah bentuk teknis untuk fragmen kata atau kalimat yang digunakan dan ditempatkan bersama pada output akhir. Terminal biasanya disebut kata atau frase.
b. Variabel
   Variabel digunakan sebagai versi abstrak dari kesimpulan yang dibuat kemudian. Prinsipnya sama dengan variabel yang digunakan dalam bahasa pemrograman atau aljabar yang sifatnya berubah-ubah. Biasanya ditulis dengan cetak tebal **variabel.**
c. Produksi
   Produksi menjelaskan bagaimana variabel dapat dikonversikan ke dalam himpunan variabel atau terminal yang berbeda. Format produksi sebagai berikut :
   variabel → kata || frase
   arti format tersebut adalah **variabel** dapat dikonversi (→) menjadi kata atau (||) frase.

2.3 Konversi Data ke *Grammar*

Contoh *grammar* sederhana dapat dijelaskan sebagai berikut :

Start → **KB** **KK** (produksi)
KB → Perusahaan kami || Wirausaha kami
KK → menawarkan Produk A || memberikan Pelayanan Ekstra
Keterangan :
KB = Kata Benda
KK = Kata Kerja

Konversi dimulai dengan Variabel **Start** kemudian dirubah menjadi variabel yang berbeda, *grammar* yang didapatkan antara lain "Perusahaan kami menawarkan produk A", " Perusahaan kami memberikan Pelayanan Ekstra", "Wirausaha kami menawarkan produk A", dan " Wirausaha kami memberikan Pelayanan Ekstra".

Contoh di atas dapat dikembangkan menjadi lebih kompleks yaitu :

Start → **KB** **KK** (produksi)
KB → Perusahaan kami || Wirausaha kami
KK → menawarkan Produk A **kapan** || memberikan Pelayanan Ekstra **kapan**
kapan → setiap bulan **apa** || kecuali bulan **apa**
apa → Mei || Desember

Setiap variabel memiliki dua pilihan, misalkan pilihan tersebut dipresentasikan dengan bit 0 dan 1. Informasi rahasia berupa bit yang ingin disembunyikan dalam *grammar* yaitu 1010. Maka rangkaian bit 1010 tersembunyi dalam kalimat "Wirausaha kami menawarkan Produk A kecuali bulan Mei". Rangkaian bit 1101 akan menghasilkan kalimat "Wirausaha kami memberikan Pelayanan Ekstra setiap bulan Desember". Jumlah semua kemungkinan kalimat yang dihasilkan adalah 2^4. Dari penjelasan tersebut dapat dijelaskan bahwa informasi yang sesungguhnya bukanlah rangkaian kalimat yang dimengerti kebanyakan orang melainkan rangkaian bit tertentu.

Tabel 1. Langkah-langkah mengkonversi 1010 menjadi sebuah kalimat

Ke-	Kalimat	Bit	Pilihan Produksi
1.	**Start**	-	**Start→KB KK**
2.	**KB KK**	1	**KB** → Wirausaha kami
3.	Wirausaha kami **KK**	0	**KK** → menawarkan Produk A
4.	Wirausaha kami menawarkan Produk A **kapan**	1	**kapan** → kecuali bulan **apa**
5.	Wirausaha kami menawarkan Produk A kecuali bulan **apa**	0	**apa** → Mei
6.	Wirausaha kami menawarkan Produk A kecuali bulan Mei	-	-

2.4 Parsing *Grammar* menjadi Data

Pelaksanaan parsing *grammar* menjadi data tergantung dari produksi yang digunakan. Ada dua faktor agar dapat melakukan parsing dengan benar yaitu :

a. Pastikan *grammar* tidak ambigu
   Jika terdapat dua kalimat yang sama didapatkan dari dua produksi *grammar* yang berbeda, maka *grammar* tersebut dikatakan ambigu. Hal ini membuat *grammar* tidak dapat menyembunyikan informasi karena tidak ada cara untuk mengembalikan menjadi data aslinya.
b. *Grammar* sesuai dengan *Greibach Normal Form* (GNF)
   Suatu CFG dikatakan berada dalam GNF apabila variabel-variabelnya berada di akhir dari produksi. Contohnya di atas sudah memenuhi GNF karena semua variabel berada di akhir produksi. Misalkan diketahui CFG "di **arah** Kota" tidak berada dalam GNF karena variabel berada di tengah produksi.

Misalkan diketahui *grammar* "Wirausaha kami memberikan Pelayanan Ekstra setiap bulan Desember", maka sesuai dengan GNF dapat dipisahkan menjadi :

Start → **KB** **KK** (produksi)
KB → Perusahaan kami || Wirausaha kami
KK → menawarkan Produk A **kapan** || memberikan Pelayanan Ekstra **kapan**
kapan → setiap bulan **apa** || kecuali bulan **apa**
apa → Mei || Desember

Tabel 2. Langkah-langkah mengkonversi kalimat menjadi bit 1101

Ke-	Fragmen Kalimat	Pilihan Produksi	Bit
1	*Wirausaha kami* memberikan Pelayanan Ekstra setiap bulan Desember	**KB**→Perusahaan kami || Wirausaha kami	1
2	Wirausaha kami *memberikan Pelayanan Ekstra* setiap bulan Desember	menawarkan Produk A **kapan** || memberikan Pelayanan Ekstra **kapan**	1
3	Wirausaha kami memberikan Pelayanan Ekstra *setiap bulan* Desember	**kapan** → setiap bulan **apa** || kecuali bulan **apa**	0
v4	Wirausaha kami memberikan Pelayanan Ekstra setiap bulan *Desember*	**apa** → Mei || Desember	1

Dari langkah-langkah di atas maka dihasilkan rangkaian bit 1101 dari *grammar*.

Implementasi *mimic function* dapat dikembangkan lebih kompleks lagi tergantung dari kreatifitas sehingga menghasilkan tingkat kesulitan tertentu. Peter Wayner menjelaskan bahwa *mimic function* dapat dikembangkan hingga tingkat kesulitan dalam membongkar keluarannya setara dengan memecahkan sistem kriptografi standart seperti RSA. Pengembangan yang lebih kompleks dan kreatif juga diimplementasikan dalam situs www.spammimic.com. Website ini menyediakan konversi data yang berupa informasi rahasia menjadi spam atau PGP atau bahasa Rusia. Oleh karena hasil konversi menggunakan tata bahasa asing, akan menjadi suatu tantangan apabila dapat dikembangkan sesuai dengan tata bahasa Indonesia.

2.5 Cara Menghasilkan *Grammar* Yang Baik

Ada 3 (tiga) cara atau sugesti untuk membangun *grammar* yang baik yaitu :
a. Pikirkan tentang plot dan naratif
b. Pecahkan kalimat menjadi banyak pilihan sehingga semakin banyak *grammar* maka semakin banyak data yang bisa dikonversikan.
c. Gunakan banyak variasi agar satu data bisa menghasilkan bentuk yang berbeda.

3. METODE PENELITIAN

3.1 Format Umum Spam

Format spam merupakan salah satu hal penting dalam mendesain keluaran dari *mimic function*. Pada tahap ini dibutuhkan kreatifitas dan perhitungan yang rumit agar dihasilkan keluaran dari *mimic function* yang dibangun tidak mudah untuk dibongkar seperti halnya dengan enkripsi. Format spam didesain sedemikian rupa sehingga sesuai dengan bentuk spam yang ada. Dengan kata lain spam yang dihasilkan tidak janggal jika dibandingkan dengan spam biasa. Umumnya format spam dapat digambarkan sebagai berikut :

Tabel 3. Format Spam Umum

Salam Pembuka	Hal Pembuka	Informasi	Hal Penutup	Salam Penutup
1	2	3	4	5

Setiap bagian pada format spam di atas merupakan satu produksi dimana memiliki grammar yang berbeda-beda. Dan setiap grammar memiliki beberapa variabel agar menjadi sebuah kalimat yang naratif sesuai dengan tiap bagian.

3.2 Desain Spam

Untuk meningkatkan kompleksitas pada format spam, format spam yang digunakan pada paper ini adalah mengkombinasikan unsur kriptografi di dalam bagian spam. Adapun format spam sebagai berikut :

Tabel 4. Format Spam Modifikasi

Salam Pembuka	Hal Pembuka	Informasi	Hal Penutup	Salam Penutup
1	2	3	4	5
Spam Pembuka		Spam Isi	Spam Penutup	
Spam Bayangan		Informasi	Cek Validitas Informasi	

Fungsi masing-masing bagian tersebut yaitu :

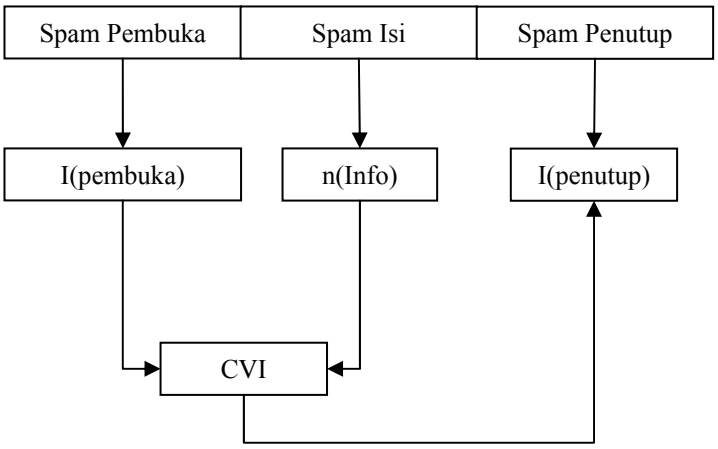
a. Spam Pembuka
Spam Pembuka merupakan spam bayangan yang berisi beberapa paragraf spam yang terdiri dari salam pembuka dan hal pembuka yang berisi perkenalan dari sebuah perusahaan atau usaha tertentu. Fungsi utama dari spam bayangan ini adalah membentuk titik awal pada pengembangan isi informasi pada spam.
Setiap ada informasi yang ingin dikonversikan ke dalam spam melalui *mimic function*, spam bayangan akan berubah-ubah. Oleh karenanya, sejumlah spam bayangan disediakan dan masing-masing diberikan nomor indeks. Nomor indeks berfungi sebagai input tambahan dalam proses pengecekan validitas informasi. Jadi spam bayangan memiliki dua fungsi seperti tersebut di atas.
Contoh spam bayangan :
"Kepada Pelanggan yang terhormat.
Kami dari perusahaan farmasi yang bergerak di bidang kesehatan sedang melakukan penelitian mengenai penyakit ... .. ."

b. Spam Isi
Spam isi merupakan hasil konversi informasi dengan *mimic function* berdasarkan produksi dan variabel tertentu yang didesain. Semakin banyak variabel dan semakin panjang produksi yang didesain maka semakin kompleks hasil keluarannya. Untuk menjaga kewajaran spam, maka bagian ini harus memiliki hubungan isi dengan spam pembuka maupun spam penutup. Dengan demikian, variabel-variabel yang digunakan pada produksi harus memilih kata-kata yang tidak ambigu dan bertentangan dengan isi.
Contoh :
" Pada kesempatan ini, kami tawarkan inovasi obat-obatan hasil penelitian kami yang bernama SUPER HERBAL. SUPER HERBAL membantu tubuh Anda menjadi lebih sehat dan meningkatkan kekebalan tubuh....."

c. Spam Penutup
Spam penutup adalah bagian cukup penting dalam keseluruhan isi spam. Spam yang dihasilkan ini adalah hasil perhitungan dari nomor indeks spam pembuka dan panjang infromasi sebelum dikonversikan ke dalam *mimic function*. Setiap spam penutup memiliki indeks yang berfungsi untuk mengecek kebenaran informasi yang diamankan atau disampaikan melalui spam. Misalkan terdapat perubahan pada isi spam maka segera diketahui bahwa spam tersebut sudah dimodifikasi sebelum orang yang berhak mendapatkan informasi itu membukanya. Proses ini disebut dengan Cek Validitas Informasi (CVI).
Perhitungan CVI dapat dijelaskan sebagai berikut :
Misalkan indeks spam pembuka dinotasikan sebagai I(pembuka), panjang informasi sebagai n(Info) dan indeks spam penutup I(penutup) maka

$$I(penutup) = CVI(I(pembuka, n(Info))$$

CVI memiliki fungsi matematika untuk menghitung masukan I(pembuka) dan n(Info) sehingga dihasilkan indeks yang unik. Indeks tersebut mengacu pada spam penutup mana yang digunakan untuk melengkapi spam pembuka dan spam isi. Proses CVI sendiri bekerja ketika mengkonversi isi spam secara keseluruhan sehingga informasi yang disembunyikan di dalamnya dapat diketahui dan diyakini tidak mengalami perubahan atau modifikasi.



Gambar 1. Proses Cek Validitas Informasi

Dari gambar di atas dapat dijelaskan bahwa apabila terjadi perubahan pada informasi maka hasil dari CVI tidak akan menghasilkan indeks spam penutup yang sama dengan indeks spam penutup pada keseluruhan spam. Demikian juga dengan indeks spam pembuka, apabila terjadi perubahan didalamnya maka spam yang diterima dinyatakan tidak valid atau sudah mengalami perubahan.

4. HASIL DAN PEMBAHASAN

4.1 Kompleksitas Hasil Keluaran

Kompleksitas dari keluaran *mimic function* dapat dilihat dari banyaknya kemungkinan kalimat yang terbentuk dari banyaknya variabel dan produksi yang digunakan. Misalkan digunakan suatu rumus produksi dengan 5 variabel dan dua pilihan yaitu bit 0 dan 1, maka banyaknya kemungkinan kalimat yang terbentuk 2^5. Apabila digunakan karakter alfabet (huruf besar dan huruf kecil), bilangan , dan beberapa karakter penting lain seperti spasi, koma dan lain sebagainya terdapat kurang lebih 75 karakter dan banyak variabel yang digunakan dalam membentuk grammar adalah 10 buah, total kemungkinan kalimat yang terbentuk 75^{10}. Banyaknya kemungkinan yang didapatkan tersebut masih menggunakan algoritma *mimic function* yang standart. Keacakan grammar yang dihasilkan pun masih kurang kompleks.

Untuk membuat suatu *grammar* acak dalam arti memiliki kompleksitas yang tinggi dan menghindari perulangan *grammar* maka diperlukan bentuk tranformasi tertentu. Seperti halnya yang terjadi dalam algoritma enkripsi dimana terdapat pola tertentu yang menyebabkan *attacker* mencari pola-pola yang sama untuk memecahkan algoritma enkripsinya, bentuk transformasi ini sangat diperlukan untuk mencegah pola-pola yang sama. Ada 3 (tiga) bentuk transformasi untuk mengacak *grammar* agar tidak terbentuk suatu pola atau perulangan yaitu :

a. Ekspansi (*Expansion*)
   Ekspansi yaitu satu variabel dalam satu produksi di-ekspansi menjadi segala kemungkinan cara di produksi lainnya. Analogi dalam aljabar seperti distribusi.
b. Kontraksi (*Contraction*)
   Kontraksi adalah kebalikan dari ekspansi yaitu jika terdapat beberapa pola dalam beberapa produksi maka diganti menjadi satu variabel yang baru.
c. Permutasi (*Permutation*)
   Permutasi adalah merubah urutan produksi.

Kombinasi dari ketiga cara di atas akan menghasilkan *grammar* baru dan menghasilkan aturan yang berbeda. Hal ini akan meningkatkan keamanan *grammar*.

4.2 Efisiensi Spam

Secara umum hasil konversi informasi yang berupa teks menjadi spam memiliki jumlah karakter yang lebih besar dibandingkan teks aslinya. Hal ini terjadi karena konversi dilakukan adalah merubah satu huruf/karakter menjadi satu kata atau beberapa kata. Hasil keluaran juga tergantung dari produksi dan variabel yang digunakan. Spam yang merupakan hasil dari produksi *mimic function* memiliki panjang kata yang sangat banyak dikatakan tidak efiseien dan tidak acak.

Spam yang baik dan acak akan menghasilkan tingkat keamanan yang lebih kuat lagi dengan memperhatikan hal-hal berikut :

a. Permasalahan yang sering timbul dalam *mimic function* adalah ketidakefisienan *grammar* dalam mengkonversi data menjadi string yang lebih panjang. Untuk menghindari hal tersebut perlu digunakan teknik algoritma enkripsi yang efisien yang menghasilkan barisan data biner tanpa menambah ukuran file.
b. Digunakan *pseudo-random number generator* dengan input kunci agar dihasilkan pola kalimat yang acak sesuai tata bahasa.

4.3 Pengembangan Selanjutnya (*future works*)

Pengamanan informasi tidak hanya diterapkan di dalam jaringan internet namun juga untuk komunikasi yang lain. Demikian pula dengan komunikasi bergerak yang saat ini semakin marak dengan banyaknya penggunaan telepon seluler. Pengembangan selanjutnya implementasi *mimic function* dalam mengamankan informasi adalah mengkonversi SMS biasa menjadi SMS promosi. Berdasarkan pengalaman penulis, SMS promosi sering kali masuk ke dalam inbox SMS. Isinya bermacam-macam antara lain mengenai ramalan astro, game, ataupun undian dengan mengirimkan sms ketik REG dan seterusnya. Bagi kebanyakan orang, SMS promosi sama halnya seperti spam di email karena kurang penting bagi pengguna telepon selular. Dengan konsep yang sama yaitu menyembunyikan informasi ke dalam spam, dapat pula diterapkan pada aplikasi mobile phone berbasis java.
Contoh yang menarik misalkan terdapat SMS berbunyi, "qt jd kmpul jam stngh smbln pg" menjadi "Pulsa

GRATIS 50.000 buat Kamu! Buruan Kirim REG HURA7 ke 3433. Utk 40 org tercepat! Kesempatan kamu utk dptin PS 3 & Liburan berdua ke Disneyland Hongkong!"

5. KESIMPULAN

Spam dapat digunakan untuk menyembunyikan informasi rahasia/penting dalam bentuk teks melalui media email. Proses konversi informasi dalam bentuk teks menjadi spam menggunakan *Mimic functio* dimana digunakan perumusan grammar yang disebut produksi dan beberapa variabel. .*Mimic function* yang merupakan salah satu teknik steganografi terbukti menggunakan unsur-unsur kriptografis didalamnya sehingga memiliki kekuatan keamanan yang tinggi bahkan setara dengan memecahkan RSA. Hasil keluaran *mimic function* yang berupa spam harus efisien, memiliki grammar yang acak untuk menghindari pola-pola kalimat yang dapat menjadi kelemahan dalam membongkat informasi yang disembunyikan.

6. DAFTAR PUSTAKA

Wayner, Peter, *Dissapearing Cryptography Information Hiding : Steganography and Watermaking*, Morgan Kaufmann Publishers, 2002.

Wayner, Peter, *Mimic functions and Tractability*, Februari 2008.

http://en.wikipedia.org/wiki/Mimic_function

http://www.spammimic.com

Soal Spam, Indonesia Tak Berdaya ,http://www.detikinet.com,18-03-2008