Rancangan Protokol dan Implementasi *Website checker* Berbasis Tanda Tangan Digital

Daniel Widya Suryanata / 13509083
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
danielsuryanata@gmail.com

***Abstract*—Makalah ini membahas tentang rancangan protocol dari *website checker*. *Website checker* adalah suatu *website* yang bertugas untuk memeriksa kevalidan dari setiap dokumen pada suatu *website*. Kevalidan tersebut dilihat dari keaslian *file-file* di dalam *website* itu, misal *file* HTML, CSS, PHP, bahkan *file* gambar yang telah ditandatangani oleh penyedia *website* tersebut. Makalah ini juga membahas implementasi dari *website checker* tersebut.**

***Index Terms*—Tanda tangan digital, *website checker*, Elgamal, SHA.**

I. PENDAHULUAN

Menurut survei yang diadakan oleh www.netcraft.com, ada sekitar 662,959,946 buah *website* yang ada di dunia per Mei 2012. Jumlah ini diperkirakan akan terus bertambah seiring dengan populernya IPv6. Tidak dapat dipungkiri bahwa perkembangan *website* yang sangat cepat tersebut membawa banyak sekali dampak positif seperti mudahnya pengguna internet untuk mengakses bahan-bahan yang disediakan oleh *website* tersebut serta berkembang pesatnya hiburan pada *website* seperti jejaring sosial, *web based game*, sampai forum-forum. Tetapi dengan bertambahnya jumlah *website*, bertambah pula ancaman akan keamanan pengguna.

Salah satu ancaman akan keamanan *website* adalah *cracking*. *Defacing* juga merupakan jenis *cracking*, namun *defacing* lebih bersifat untuk mengubah antarmuka *website* semata, sedangkan *cracking* lebih bertujuan untuk mengacaukan sistem yang ada pada *website* tersebut. *Cracking* atau yang biasa disebut dengan hacking (*black hat hacking*) cukup merepotkan karena kadang baik pengguna maupun pengembang dari *website* itu sendiri tidak menyadari bahwa *website* tersebut menjadi korban *cracking*. *Cracking* dilakukan dengan mengubah kode dari *file* di *website* tersebut, misalnya mengubah sintaks HTML, PHP, Javascript.

Salah satu cara mencegah bahaya *hacking* adalah dengan menggunakan *website checker*. *Website checker* adalah suatu *website* yang digunakan untuk memeriksa kevalidan dokumen-dokumen yang membentuk suatu *website* seperti *file* HTML, CSS, PHP, script javascript, bahkan *file* gambar sekalipun.

Prinsip dari *website checker* pada makalah ini adalah pengembang *website* memberikan seluruh *file* dari *website*nya pada *website checker*. *Website checker* akan menyimpan *file* asli tersebut. Setiap kali ada *request* dari pengguna terhadap *website* tersebut, *website checker* memeriksa seluruh *file* yang ada pada *website* tersebut. Apabila *file* tersebut sama, maka artinya *website* tersebut aman untuk digunakan.

Gambar 1. Ilustrasi cracking

II. KRIPTOGRAFI DALAM *WEBSITE CHECKER*

Website checker ini sangat bergantung pada prinsip-prinsip kriptografi. Berikut ini adalah penjelasan singkat mengenai prinsip-prinsip tersebut:

A. Kriptografi Kunci Publik

Kriptografi kunci publik (kriptografi asimetri) ditemukan pada tahun 1876. Makalah pertama yang membahas mengenai kriptografi kunci public ini adalah makalah dari Diffie-Hellman yang berjudul "*New Directions in Cryptography"* yang selanjutnya akan dikenal dengan metode pertukaran kunci Diffie-Hellman.

Dalam kriptografi kunci publik, seorang memiliki dua kunci untuk berkomunikasi dengan orang lain, yaitu sebuah kunci publik dan sebuah kunci privat. Kunci publik digunakan untuk

mengenkripsi pesan, sedangkan kunci privat digunakan untuk mendekripsi pesan terenkripsi tersebut.

Kriptografi kunci publik memiliki beberapa keunggulan. Salah satunya adalah sulitnya untuk memecahkan (melakukan kriptanalisis) pesan terenkripsi karena kekuatan enkripsi dan dekripsi pada kriptografi kunci publik bersumber pada sulitnya memecahkan masalah pemfaktoran dan logaritma diskrit.

Keunggulan lain dari kriptografi ini adalah jumlah kunci yang dapat ditekan. Apabila pada suatu perkumpulan, setiap anggota dari perkumpulan itu ingin berkomunikasi secara rahasia dengan anggota lain, maka anggota tersebut harus menyetujui sebuah kunci rahasia dengan setiap anggota lainnya. Maka dengan skema kunci rahasia ini, di dalam organisasi tersebut harus ada sebanyak N x (N - 1) / 2 jumlah kunci, dengan N adalah banyaknya orang di dalam organisasi tersebut.

Sedangkan dengan kriptografi kunci publik, hanya dibutuhkan sebanyak 2N kunci dengan N adalah banyaknya orang di dalam organisasi tersebut. Seorang anggota cukup menyimpan kunci privatnya dan menyebarkan kunci publiknya untuk dapat berkomunikasi secara rahasia dengan anggota yang lain. Dengan kriptografi kunci publik, jumlah kunci yang ada dapat ditekan.

B. Fungsi Hash

Fungsi hash adalah suatu fungsi yang menerima masukan *String* dengan panjang sembarang dan mengeluarkan / mentransfor-masukannya menjadi sebuah *String* dengan panjang yang tetap. Umumnya panjang keluaran tersebut lebih kecil dibandingkan panjang *String* masukan.

Fungsi hash dapat didefinisikan sebagai berikut :

$$h = H(m)$$

Dimana m adalah panjang pesan dan h adalah String yang dihasilkan. h biasanya akan jauh lebih kecil dibandingkan m.

Berbeda dengan enkripsi, hash memiliki keunikan yaitu *String* yang dihasilkan (h) tidak dapat dikembalikan menjadi pesan asli (m), sehingga fungsi hash sering disebut *one way hash*. Dalam implementasinya, ada beberapa fungsi hash yang sering dipakai, diantaranya adalah MD5 dan SHA. Namun dalam makalah ini, fungsi hash yang akan digunakan adalah SHA-1.

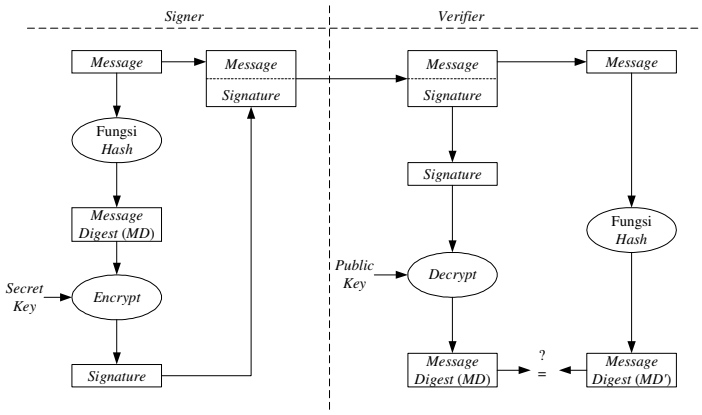
Kegunaan utama dari fungsi hash ada tiga, yaitu untuk menjaga integritas data, menghemat waktu pengiriman, dan menormalkan panjang data yang beraneka ragam.

Collision atau tabrakan bisa saja terjadi pada fungsi hash. *Collision* disebabkan oleh dua atau lebih pesan m yang apabila dihash akan menghasilkan nilai yang sama. Contoh fungsi hash yang dapat menghasilkan *collision* adalah SHA-0. Tetapi untuk keluarga SHA yang lain seperti SHA-1 dan SHA-512, *collision* sudah dapat dihindari.

C. Tanda Tangan Digital

Banyak orang salah kaprah mengenai pengertian dari tanda tangan digital. Tanda tangan digital atau yang lebih familiar dikenal dengan *digital signature* bukanlah tanda tangan tertulis yang discan, melainkan nilai kriptografis yang bergantung pada isi pesan dan kunci sehingga tanda tangan digital untuk pesan yang berbeda akan berbeda pula.

Tanda tangan digital dapat dibuat dengan menggunakan kriptografi simetri maupun asimetri. Namun tanda tangan digital menggunakan kriptografi asimetri memiliki keuntungan karena memberikan aspek keamanan nir-penyangkalan.



Gambar 2. Skema Tanda Tangan Digital dengan kriptografi asimetri

D. Algoritma Elgamal

Algoritma Elgamal adalah algoritma kriptografi asimetri yang diciptakan oleh Taher Elgamal pada tahun 1985 dalam makalahnya yang berjudul "*A public key cryptosystem and a signature scheme based on discrete logarithms*".

Keamanan dari algoritma ini terletak pada sulitnya menghitung logaritma diskrit, yaitu jika p adalah bilangan prima dan g dan y adalah sembarang bilangan bulat, carilah x sedemikian sehingga :

$$g^x = y \pmod p$$

Algoritma Elgamal memfasilitasi kebutuhan untuk berkirim pesan secara rahasia (mengenkrip pesan dengan kunci publik dan mendekripnya dengan kunci privat) maupun untuk kebutuhan tanda tangan digital (mengenkrip pesan dengan kunci privat dan memverifikasinya dngan kunci publik). Dalam makalah ini akan digunakan algoritma Elgamal untuk tanda tangan digital.

Dalam algoritma ini, seorang pengguna akan memiliki kunci publik dan kunci privat. Kunci privat adalah suatu bilangan yang disebut x. Kunci publik adalah kombinasi dari bilangan prima p, bilangan a, dan bilangan y, dimana:

$$y = a^x \bmod p$$

Kemudian satu bilangan *random* k dipilih. k harus memenuhi GCD(k, p-1) = 1, atau dengan kata lain, k dan p-1 harus relatif prima.

Berikutnya hitung K dengan rumus:

$$K = a^k \pmod p$$

Dengan menggunakan algoritma *extended Euclidean* (*inverse*), S dapat ditemukan dengan rumus:

$$M = x.K + k.S \bmod (p-1)$$

Atau

$$S = k^{-1}(M - x.K) \bmod (p-1)$$

Dengan M adalah pesan yang ingin dienkripsi, dalam implementasi makalah ini, M adalah nilai hash suatu dokumen.

Sehingga didapatkan tanda tangan digitalnya adalah (K,S). Setelah dipakai, k harus dihancurkan / dibuang.

Sedangkan untuk memverifikasi kode tersebut, seseorang hanya perlu untuk memastikan bahwa:

$$y^K.K^S \bmod p = a^M \bmod p$$

Apabila ruas kiri sama dengan ruas kanan, berarti tanda tangan digital itu valid.

III. MEKANISME *WEBSITE CHECKER*

Pada bab ini akan dibahas mengenai berbagai mekanisme yang ada pada *website checker*.

A. Rancangan Protokol

Pada protokol ini akan ada tiga pihak yang terlibat, yaitu pihak penyedia jasa *website*, konsumen(pembaca), serta pihak *website checker* itu sendiri.

Berikut ini adalah langkah-langkah dari protokol *website checker*:

1. Kunci publik dan kunci privat dibangkitkan oleh jasa penyedia *website* untuk dirinya sendiri.
2. Setiap *file* yang ada di *website*nya, seperti index.php, *file* HTML, *file* CSS, *file* PHP, serta *file* gambar dihash dengan menggunakan algoritma SHA-1.
3. Setelah setiap *file* itu dihash, tanda tangan digital dapat dibangkitkan / dibuat dengan menggunakan algoritma Elgamal untuk tanda tangan digital.
4. Nilai dari setiap tanda tangan digital tersebut disimpan.
5. Kunci publik dari jasa penyedia *website* dikirimkan kepada *website checker*. Selanjutnya *website checker* akan menyimpan pasangan nama *website* dan kunci publiknya dalam basis datanya.
6. Nilai hash dari masing-masing *file* yang ada di dalam *website* juga akan dikirimkan kepada *website checker* sehingga nantinya *website checker* akan menyimpan pasangan nama *website*, nama *file*, dan nilai hash *file* tersebut di dalam basis datanya.
7. Apabila di kemudian waktu, konten dari *website* tersebut ingin diubah oleh pihak penyedia *website*, maka pihak penyedia *website* harus menghitung tanda tangan digital yang baru dari *file* tersebut kemudian mengirimkannya lagi kepada *website checker*.
8. Apabila konten suatu *website* hendak diakses oleh pengguna, maka pengguna tersebut menghubungi *website checker* terlebih dahulu.
9. Setelah mendapat *request*, *website checker*-lah yang akan mengakses *website* yang diminta tersebut. *Website checker* akan memeriksa *file* dalam *website* yang hendak diakses oleh pengguna. *Website checker* memeriksa dengan melakukan hash pada konten *file*, kemudian memverifikasinya dengan algoritma Elgamal untuk verifikasi. Untuk verifikasi ini dibutuhkan kunci publik pihak penyedia *website* yang sebelumnya telah disimpan di dalam basis data *website checker*.
10. Apabila *file* tersebut valid, maka *website checker* akan mengizinkan pengguna untuk mengakses *website* tersebut. Tetapi apabila *file* tersebut tidak valid, maka *website checker* akan memberikan peringatan kepada pengguna bahwa konten *website* tersebut mungkin telah diubah oleh pihak ketiga, namun *website checker* tidak akan melarang pengguna untuk membuka *website* tersebut.
11. Untuk meningkatkan keamanan, disarankan bagi pihak penyedia *website* untuk mengubah kunci publik dan kunci privat mereka secara berkala dan mengirimkannya kedua kunci tersebut serta nilai hash baru yang telah dihash dengan menggunakan kunci tersebut kepada *website checker*.

B. Aspek Keamanan

Dengan menggunakan *website checker*, aspek keamanan pengguna akan tercapai, dengan asumsi pihak penyedia jasa *website* tidak memberikan kode yang berbahaya ataupun *bug* pada *file-file* di dalam *website*nya. Aspek keamanan hanya meliputi keamanan *file* dari pihak ketiga, atau *cracker*.

Sebagai contoh, anggap *source code* dibawah ini sebagai source code asli dari pihak penyedia jasa *website*:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
```

```
<head>
<title>Welcome</title>
<meta http-equiv="REFRESH" content="0;url=http://www.perjaka-cerdas.com"></HEAD>
<BODY>
    <p> Welcome to my first website </p>
</BODY>
</HTML>
```

Misalkan ada seorang *cracker* yang ingin mengubah *redirect page* ke tempat lain dari *source code* diatas:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Welcome</title>
<meta http-equiv="REFRESH" content="0;url=http://www.perawan-rajin.com"></HEAD>
<BODY>
    <p> Welcome to my first website </p>
</BODY>
</HTML>
```

Dengan menggunakan fungsi hash SHA-1, *source code* asli akan menghasilkan nilai hash: db29ff3e8b0d7d4cb88c672d99e4101881a64419. Sedangkan dengan fungsi yang sama, *source code* yang sudah diubah oleh *cracker* akan menghasilkan nilai hash: 8b53e779c51897916e6be4886d6a715be1417ea9.

Salah satu keunggulan dari fungsi hash adalah perubahan sekecil apapun yang terjadi akan menghasilkan nilai hash yang jauh berbeda dengan aslinya, sehingga bagi manusia sekalipun, mudah sekali untuk mengamati perubahan pada nilai hash tersebut.

C. Aspek Nir-penyangkalan

Seringkali memenuhi aspek keamanan saja tidaklah cukup. Walau pengguna sudah memeriksakan *website* yang ingin dikunjunginya melewati *website checker*, terkadang masih terjadi hal-hal yang membahayakan seperti adanya bug ataupun kode berbahaya lainnya. Dalam menyikapi hal ini, seringkali pihak penyedia jasa *website* mengelak untuk bertanggung jawab walau tidak ada bukti adanya intervensi pihak ketiga.

Untuk menyelesaikan masalah tersebut, aspek nir-penyangkalan menjadi sangat penting. Nir-penyangkalan adalah suatu jaminan bahwa seseorang tidak bisa menolak bahwa ia telah melakukan sesuatu. Pada makalah ini, aspek nir-penyangkalan adalah dengan menjamin bahwa pihak penyedia tidak dapat menyangkal bahwa perubahan yang terjadi pada *file*nya bukan karena pihak ketiga.

Apabila *website checker* sudah menginformasikan kepada pengguna bahwa *website* yang ingin dikunjungi aman, tetapi ternyata *website* tersebut tidak aman, pengguna dapat menuntut pihak penyedia jasa *website*, dan pihak penyedia jasa *website* tidak akan dapat menyalahkan pihak ketiga, karena apabila ada intervensi pihak ketiga, maka hal tersebut sudah ditangani di dalam aspek keamanan.

IV. Implementasi

Website checker ini telah diimplementasikan secara sangat sederhana. Berikut adalah implementasinya:

A. Skema Basis Data

Pada *website checker* ini ada dua tabel yang digunakan. Tabel pertama adalah tabel bernama kuncipublik yang memiliki empat kolom yaitu Nama_Website, P, A, dan Y. *Primary key* pada tabel ini adalah Nama_Website. Sedangkan P, A, Y adalah kolom untuk menampung sebuah kunci publik dari pihak penyedia jasa *website*. Seluruh kolom pada tabel ini bertipe varchar dengan 40 karakter.

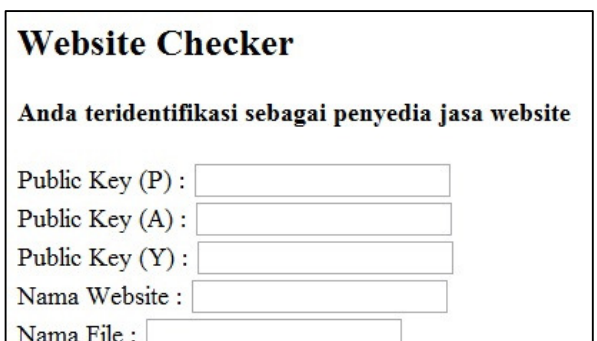
Tabel kedua adalah tabel bernama hash*file* yang memiliki tiga kolom yaitu Nama_Website, Nama_File, dan Nilai_Hash. *Primary key* pada tabel ini adalah gabungan Nama_Website dan Nama_File. Seluruh kolom pada tabel ini bertipe varchar dengan 40 karakter, kecuali kolom Nilai_Hash yang bertipe varchar dengan 100 karakter. Kolom Nilai_Hash memiliki penampung yang lebih besar karena harus menampung tanda tangan digital yang berupa kombinasi R dan S.

B. Implementasi pada Pihak Penyedia Jasa Website

Secara garis besar, *website checker* memiliki dua fungsionalitas utama, yaitu implementasi pada pihak penyedia jasa *website* dan implementasi pada pihak pengguna.

Antarmuka *website checker* ini masih tergolong sangat sederhana untuk memudahkan pengguna dalam navigasi, bahkan untuk pengguna pemula sekalipun. Fung-sionalitas dalam *website* ini dapat dibedakan dan dilihat secara jelas mulai dari halaman utama (Gambar 3).

Gambar 3. Halaman Utama *Website checker*



Gambar 4. Halaman untuk Penyedia Jasa *Website*

Untuk penyedia jasa *website*, halamannya didesain sedemikian rupa untuk memudahkan memasukkan nilai. Yang perlu dimasukkan adalah tiga parameter kunci publik, yaitu P, A, dan Y, nama *website*, nama *file*, dan nilai SHA-1 dari *File*. Apabila pihak penyedia jasa *website* baru pertama kali menggunakan *website checker*, maka pihak penyedia wajib memasukkan ketiga parameter kunci publik. Apabila pihak penyedia ingin mengubah kunci publiknya, pihak penyedia hanya perlu memasukkan nilai baru ke dalam ketiga kolom kunci publik tersebut.

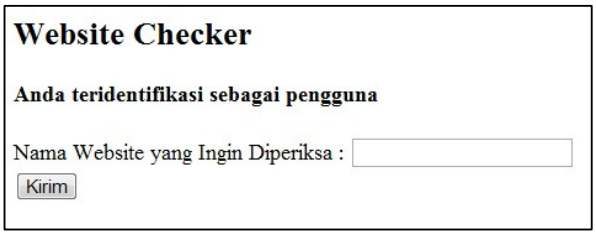
Kolom Nama Website, Nama File, dan Nilai SHA-1 dari *File* wajib diisi oleh pihak penyedia jasa *website*. Namun untuk memasukkan nilai hash setiap *file*, *form* pada Gambar 4 masih harus diisikan satu per satu.

Apabila tombol kirim ditekan, maka seluruh nilai tersebut akan masuk ataupun terupdate ke dalam basis data. Kolom Nama *Website*, P, A, dan Y akan masuk ke dalam tabel kuncipublik. Sedangkan kolom Nama Website, Nama File, dan Nilai SHA-1 dari *File* akan masuk ke dalam tabel hashfile.

Program untuk menghitung tanda tangan digital belum diintegrasikan dengan *website checker* sehingga harus menggunakan program yang terpisah.

C. Implementasi pada Pihak Pengguna

Pada halaman pengguna, form yang harus diisi akan jauh lebih sederhana. Pengguna hanya perlu mengisikan nama *website* yang akan diperiksa. Selanjutnya *website checker* akan melakukan hash pada seluruh *file* dalam *website* yang akan diperiksa tersebut. Kemudian *website checker* akan membandingkan nilainya dengan nilai tanda tangan digital yang sudah tersimpan dalam tabel hashfile.



Gambar 5. Halaman untuk Pengguna

Apabila tombol 'Kirim' ditekan maka *website checker* akan memeriksa seluruh *file* yang ada di dalam *website* tersebut. Kemudian akan mengeluarkan halaman baru yang berisi valid atau tidaknya halaman tersebut.

V. CONTOH PENGGUNAAN *WEBSITE CHECKER*

Misalkan ada pihak penyedia jasa *website* yang memiliki *website* bernama www.ilmucoding.com ingin memasukkan *website*nya ke dalam daftar *website checker*. *Website* tersebut memiliki 3 *file*, yaitu *file* index.php, artikel1.html, dan artikel2.

Kunci privat dan publik dari penyedia tersebut adalah:
x:917124101247123912271239122301481238122401239185122138124712 38
p: 18213114123012398124120981247
a: 81283129814712938124981248
y: 554209137417435170741780326 7

Apabila index.php ditanda tangani dengan menggunakan kedua kunci tersebut, maka hasilnya:

1a5250fa69dc30191ea03698,1a7f02d59f81563db9c4c5e6

Bila artikel1.html ditanda tangani, maka hasilnya:

595620a2fc6ab84f5b47aef,df0e5f951921558a1068c11

Bila artikel2.html ditanda tangani, maka hasilnya:

39dbdab2a41c031b410b023,1f85adb9595074c417613e3d

Misalkan index.php diubah sedikit. Dari judul "ilmu coding" diubah menjadi "ilmu codang", maka hasil tanda tangan digitalnya:

2fd6ee901d307adf39fc4c64,37173d02f8a9ac28692baa6e

Hasil tersebut sangat berbeda dengan tanda tangan digital *file* asli, sehingga *website checker* akan mengembalikan pesan kesalahan.

VI. KESIMPULAN

Website checker dengan menggunakan tanda tangan digital(kriptografi asimetri) memiliki keuntungan dalam menjamin aspek keamanan dan aspek nir-penyangkalan.

Untuk meningkatkan keamanan, hendaknya pihak penyedia jasa *website* rajin dalam mengganti kunci dan rajin mengirim update tanda tangan digital dari *file-file* yang ada di *website*nya. Pengguna juga hendaknya rajin dalam menggunakan *website checker* sebelum hendak mengakses suatu *website*.

Belum banyak pihak yang mengembangkan *website checker*. Hal ini mungkin disebabkan karena sulitnya implementasi dan biaya dari *website checker* itu sendiri.

Website checker yang penulis buat belum sempurna seluruhnya, masih banyak kekurangan karena penulis lebih menekankan pada rancangan protokol. Berbagai

kemungkinan perbaikan dalam implementasi maupun dalam rancangan protokol masih terbuka lebar.

REFERENCES

[1] http://news.netcraft.com/archives/category/web-server-survey/ W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135. Tanggal akses : 11 Mei 2012, pukul 14.00.

[2] http://courses.cs.tamu.edu/pooch/665_spring2008/Australian-sec-2006/less19.html. Tanggal akses : 12 Mei 2012, pukul 13.00.

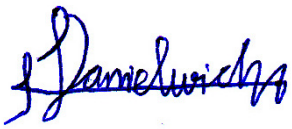
[3] http://searchsecurity.techtarget.com/definition/nonrepudiation. Tanggal akses : 12 Mei 2012, pukul 14.00.

[4] Munir, Rinaldi. 2009. "Diktat Kuliah IF3058, Kriptografi," Bandung : Program Studi Teknik Informatika ITB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 13 Mei 2012



Daniel Widya Suryanata
13509083