IMPLEMENTASI *HASH FUNCTION* DALAM MESSAGE DIGEST 5 (MD5)

Satya Fajar Pratama – NIM : 13506021

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16021@students.if.itb.ac.id

Abstrak

Dewasa ini, masalah keamanan menjadi salah satu aspek yang paling penting dalam proses pengiriman informasi, baik berupa pesan maupun data. Dalam hal ini, masalah keamanan tersebut berkaitan erat dengan kerahasiaan, keutuhan/integritas data, penghindaran penolakan (non-repuditation), dan autentikasi. Hal-hal tersebut menjadi sangat krusial supaya informasi yang dikirimkan tidak dapat dibaca oleh orang yang tidak berkepentigan, tidak mengalami perubahan sewaktu proses pengiriman, dan diterima oleh pihak yang dituju/dimaksud.
Message Digest 5 (MD5) adalah salah satu alat yang memberi garansi bahwa pesan yang dikirim akan sama dengan pesan yang diterima, hal ini dengan membandingkan 'sidik jari' atau 'intisari pesan' kedua pesan tersebut. MD5 merupakan pengembangan dari MD4 dimana terjadi penambahan satu ronde. MD5 memproses teks masukan ke dalam blok-blok bit sebanyak 512 bit, kemudian dibagi ke dalam 32 bit sub blok sebanyak 16 buah. Keluaran dari MD5 berupa 4 buah blok yang masing-masing 32 bit yang mana akan menjadi 128 bit yang biasa disebut nilai hash.

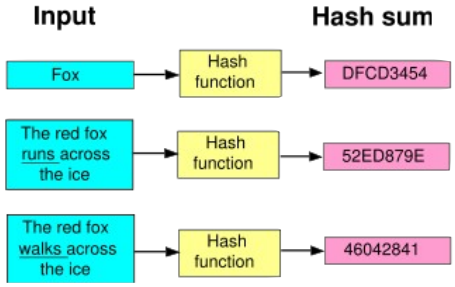
***Kata kunci** : Hash function, Kriptografi, MD5*

1. PENDAHULUAN

Hash function adalah suatu metode yang digunakan untuik mengubah data-data yang ada menjadi sebuah bilangan yang relatif kecil (*small number*) yang akan menjadi "sidik jari" (*fingerprint*) dari data terebut. Fungsi ini memecah dan mengolah data untuk menghasilkan kode atau nilai hashnya. Nilai hash dari suatu fungsi hash akan memiliki panjang yang tetap untuk masukan dengan panjang yang sembarang. Secara umum, fungsi hash memiliki beberapa sifat utama, yaitu : fungsi satu arah, artinya untuk suatu nilai fungsi hash y, sulit menemukan nilai input x yang memenuhi persamaan H(x)=y, dan *collision free/resistant*, artinya sulit untuk menemukan 2 buah nilai input yang mempunyai nilai fungsi hash yang sama.

Salah satu fungsi hash yang banyak digunakan adalah *Message Digest 5* (MD5). MD-5 merupakan fungsi hash satu arah yang diciptakan oleh Ron Rivest. MD-5 adalah salah satu aplikasi yang digunakan untuk mengetahui bahwa pesan yang dikirim tidak ada perubahan sewaktu berada di jaringan.
Algoritma MD-5 secara garis besar adalah mengambil pesan yang mempunyai panjang variable diubah menjadi 'sidik jari' atau 'intisari pesan' yang mempunyai panjang tetap yaitu 128 bit. 'Sidik jari' ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari 'sidik jari' MD-5.
Message digest atau intisari pesan harus mempunyai tiga sifat penting, yaitu :

- Bila P diketahui, maka MD(P) akan dengan mudah dapat dihitung.
- Bila MD(P) diketahui, maka tidak mungkin menghitung P.
- Tidak seorang pun dapat memberi dua pesan yang mempunyai intisari pesan yang sama. $H(M) \neq H(M')$.



2. KRIPTOGRAFI DAN FUNGSI HASH SATU ARAH

Pada bagian ini, akan diterangkan mengenai kriptografi secara umum, prinsip dasar dan tujuan dari kriptografi, fungsi hash satu arah dan juga algoritma *Message Digest 5* (MD5).

2.1. Prinsip Dasar Kriptografi

Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan (data atau informasi) dengan cara menyamarkannya (*to crypt* artinya menyamar) menjadi bentuk tersandi yang tidak mempunyai makna. Kriptografi digunakan untuk menyembunyikan informasi rahasia dari pihak yang tidak berhak membacanya. Menurut sejarahnya, kriptografi sudah sejak lama digunakan oleh tentara Sparta di Yunani pada permulaan tahun 400 SM. Mereka menggunakan alat yang disebut *scytale*. Alat ini terdiri dari sebuah pita panjang dari daun papyrus yang dililitkan pada sebatang silinder.

Kriptografi mempunyai 2 komponen utama, yaitu :

1. Plaintext : teks asli, teks jelas yang dapat dimengerti
2. Chipertext : teks tersandi, teks yang sudah mengalami pemrosesan
3. Cipher dan key, cipher adalah fungsi matematika yang digunakan untuk melakukan enkripsi dan dekripsi pada sebuah data, sedangkan key adalah sekumpulan bit yang diperlukan untuk mengenkripsi dan mendekripsi data.

Kriptografi mempunyai 2 (dua) bagian penting yaitu **enkripsi** dan **dekripsi**. Enkripsi adalah proses dari penyandian pesan asli (*Plain text*) menjadi pesan yang sudah disandikan (*Cipher text*).

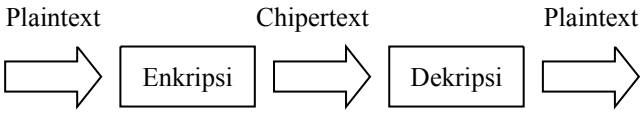
C = E (P), dimana
P = pesan asli (*Plain text*)
E = proses enkripsi (*Encryption*)
C = pesan dalam bahasa sandi (*Chiper text*)

Sedangkan, dekripsi *a*dalah proses kebalikan dari *enkripsi* yaitu mengubah pesan yang sudah disandikan (*Cipher text*) menjadi pesan asli (*Plain text*).

P = D (C), dimana
D = proses dekripsi

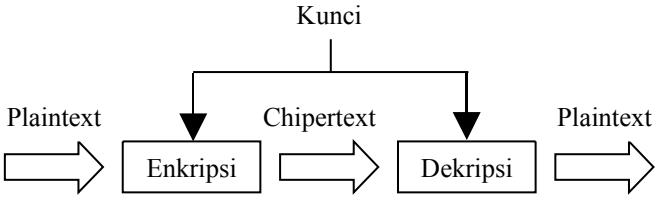
Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka kesamaan berikut haruslah terpenuhi.

P = D(E(P))



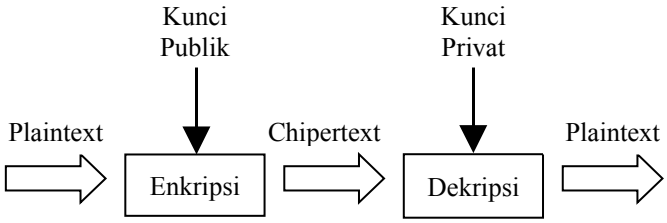
Ada 2 model algoritma enkripsi yang menggunakan kunci, yaitu kunci simetrik dan kunci asimetrik.

1. Kunci simetrik yaitu kunci yang digunakan untuk melakukan enkripsi dan dekripsi identik/sama. Kunci-kunci simetrik harus dirahasiakan. Keuntungan dari algoritma ini adalah operasi yang cepat atau hanya membutuhkan waktu yang sedikit.



Gambar 2.1.1 Kunci Simetrik

2. Kunci asimetrik yaitu kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda/tidak sama. Algoritma enkripsi kunci asimetrik juga disebut kriptografi kunci publik karena kunci untuk enkripsi dapat diketahui oleh umum (*public key*), tetapi kunci untuk dekripsi hanya boleh diketahui oleh pihak yang berwenang saja (*private key*). Keuntungan algoritma ini adalah hanya diperlukan 2 buah kunci saja, *private key* dan *public key* untuk berhubungan secara rahasia dengan banyak pihak.



Gambar 2.1.2 Kunci Asimetrik

2.2 Tujuan Kriptografi

Pada mulanya, kriptografi hanya bertujuan untuk aspek keamanan data atau informasi saja. Namun, seiring dengan berkembangnya ilmu kriptografi, tujuan dari kriptografi pun turut berkembang. Adapun tujuan kriptografi saat ini yaitu :

1. *Privasi*, Musuh tidak dapat membongkar tulisan yang kita kirim.
2. *Autentikasi*, Penerima pesan dapat meyakinkan dirinya bahwa pesan yang diterima tidak terjadi perubahan dan berasal dari orang yang diinginkan.
3. *Tanda tangan*, penerima pesan dapat meyakinkan pihak ketiga bahwa pesan yang yang diterima berasal dari orang yang diinginkan.
4. *Minimal*, Tidak ada yang dapat berkomunikasi dengan pihak lain kecuali berkomunikasi dengan pihak yang diinginkan.
5. *Pertukaran bersama,* suatu nilai (misalnya tanda tangan sebuah kontrak) tidak akan dikeluarkan sebelum nilai lainnya (misalnya tanda tangan pihak lain) diterima.

6. *Koordinasi,* di dalam komunikasi dengan banyak pihak, setiap pihak dapat berkoordinasi untuk tujuan yang sama walaupun terdapat kehadiran musuh.

2.3 Fungsi Hash Satu Arah

Fungsi hash satu arah adalah fungsi hash yang memiliki beberapa sifat keamanan tambahan sehingga dapat dipakai untuk tujuan keamanan data. Umumnya, fungsi hash satu arah ini digunakan untuk keperluan autentikasi dan integritas data.

Fungsi hash digunakan untuk menempatkan suatu *record* yang mempunyai nilai kunci *k*. Fungsi hash yang paling umum berbentuk

$$h(k) = k \bmod m$$

yang dalam hal ini *m* adalah jumlah lokasi memori yang tersedia (misalkan memori berbentuk sel-sel yang diberi indeks 0 sampai *m*-1). Fungsi *h* di atas menempatkan *record* dengan kunci *k* pada suatu lokasi memori yang beralamat *h(k)*.

Fungsi hash satu arah memiliki banyak nama, yaitu : fungsi pembanding, fungsi penyusutan, intisari pesan, sidik jari, *message integrity check* (MIC) atau pemeriksa keutuhan pesan dan *manipulation detection code* (MDC) atau pendektesi penyelewengan kode.

Fungsi hash satu arah dibuat berdasarkan ide tentang fungsi pemampatan. Fungsi hash adalah sebuah fungsi atau persamaan matematika yang mengambil input dengan panjang variabel (*preimage*) dan merubahnya menjadi panjang yang tetap (biasanya lebih pendek), keluarannya biasa disebut nilai hash.

Fungsi hash satu arah adalah sebuah fungsi hash yang berjalan hanya satu arah. Adalah mudah untuk menghitung nilai hash dari *pre-image*, tetapi sangat sulit untuk membangkitkan *pre-image* dari nilai hash-nya.

Metode fungsi hash satu arah adalah berfungsi melindungi data dari modifikasi. Apabila ingin melindungi data dari modifikasi yang tidak terdeteksi, dapat dihitung hasil fungsi hash dari data tersebut, selanjutnya dapat menghitung hasil fungsi hash lagi dan membandingkannya dengan hasil yang pertama apabila berbeda maka terjadi perubahan selama pengiriman.

Sebagai contohnya, apabila terapat seorang pengirim (A) yang akan mengirim pesan kepada temannya (B). Sebelum mengirim, A melakukan hash pada pesannya untuk mendapatkan nilai hashnya, kemudian dia mengirim pesan itu beserta nilai hashnya, Lalu B melakukan hash untuk mencari nilai hash dari pesan itu bila terjadi perbedaan maka sewaktu pengiriman telah terjadi perubahan dari pesan tersebut.

Masukan dari fungsi hash satu arah adalah blok pesan dan keluaran dari blok text atau nilai hash sebelumnya ini dapat dilihat pada Gambar 2.3.1sehingga secara garis besar, hash dari blok Mi adalah:

$$\mathbf{hi} = f(\mathbf{Mi}, \mathbf{hi - 1})$$

Nilai hash ini bersama blok pesan berikutnya menjadi masukan berikutnya bagi fungsi pemampatan. Nilai hash keseluruhan adalah nilai hash dari blok paling akhir. *Pre-image* sedapatnya mengandung beberapa binari yang menggambarkan panjang dari masukan pesan. Teknik ini digunakan untuk mengatasi masalah yang dapat terjadi bila pesan yang mempunyai pesan yang tidak sama mempunyai nilai hash yang sama. Metode ini biasa disebut **MD-*strengthening*** atau **penguatan MD.**

Gambar 2.3.1 Fungsi Hash Satu Arah

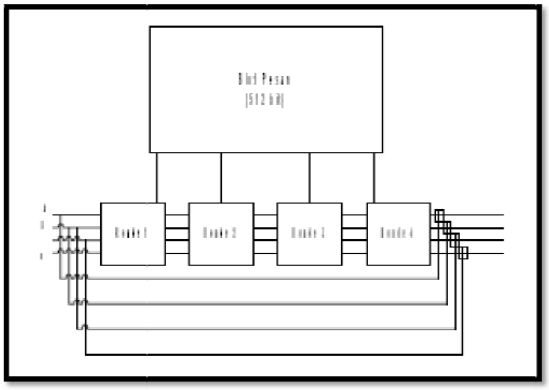
2.4 *Message Digest 5* (MD5)

Bagian ini akan menjelaskan secara spesifik sistem kriptografi MD5, yaitu algoritma MD5 mulai dari masukan (*input*) hingga keluarannya (*output*).

2.4.1 Prinsip Dasar MD5

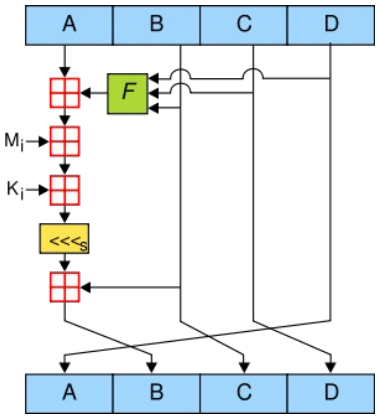
Message Digest 5 (MD-5) adalah salah satu penggunaan fungsi hash satu arah yang paling banyak digunakan. MD-5 merupakan fungsi hash kelima yang dirancang oleh Ron Rivest. MD-5 merupakan pengembangan dari MD-4 dimana terjadi penambahan satu ronde. MD-5 memproses teks masukan ke dalam blok-blok bit sebanyak 512 bit, kemudian dibagi ke dalam 32 bit sub blok sebanyak 16 buah. Keluaran dari MD-5 berupa 4 buah blok yang masing-masing 32 bit yang mana akan menjadi 128 bit yang biasa disebut nilai hash.

Pada Gambar 2.4.1.1 terlihat simpul utama dari MD-5. Simpul utama MD5 mempunyai blok pesan dengan panjang 512 bit yang masuk ke dalam 4 buah ronde. Hasil keluaran dari MD-5 adalah berupa 128 bit dari byte terendah A dan tertinggi byte D.



Gambar 2.4.1.1 Simpul Utama MD5

2.4.2 Penjelasan Algoritma MD-5



Gambar 2.4.2.1 Satu operasi MD5 — MD5 terdiri atas 64 operasi, dikelompokkan dalam empat putaran dari 16 operasi. *F* adalah fungsi nonlinear; satu fungsi digunakan pada tiap-tiap putaran. *Mi* menujukkan blok 32-bit dari masukan pesan, dan *Ki* menunjukkan konstanta 32-bit, berbeda untuk tiap-tiap operasi. $<<<_s$ menunjukkan perputaran bit kiri oleh s, s bervariasi untuk tiap-tiap operasi. ⊞ menunjukkan tambahan modulo 2^{32}.

Setiap pesan yang akan dienkripsi, terlebih dahulu dicari berapa banyak bit yang terdapat pada pesan. Kita anggap sebanyak b bit. Di sini b adalah bit non negatif integer, b bisa saja nol dan tidak harus selalu kelipatan delapan. Pesan dengan panjang b bit dapat digambarkan seperti berikut :

m_0 m_1 …..m_(b-1)

Terdapat 5 langkah yang dibutuhkan untuk untuk menghitung intisari pesan. Adapun langkah–langkah tersebut dijelaskan pada subbab-subbab berikut.

2.4.2.1 Menambahkan bit

Pesan akan ditambahkan bit-bit tambahan sehingga panjang bit akan kongruen dengan 448, mod 512. Hal ini berarti pesan akan mempunyai panjang yang hanya kurang 64 bit dari kelipatan 512 bit. Penambahan bit selalu dilakukan walaupun panjang dari pesan sudah kongruen dengan 448, mod 512 bit. Penambahan bit dilakukan dengan menambahkan "1" di awal dan diikuti "0" sebanyak yang diperlukan sehingga panjang pesan akan kongruen dengan 448, mod 512.

2.4.2.2 Penambahan Panjang Pesan

Setelah penambahan bit, pesan masih membutuhkan 64 bit agar kongruen dengan kelipatan 512 bit. 64 bit tersebut merupakan perwakilan dari b (panjang pesan sebelum penambahan bit dilakukan). Bit-bit ini ditambahkan ke dalam dua word (32 bit) dan ditambahkan dengan *low-order* terlebih dahulu. Penambahan pesan ini biasa disebut juga ***MD Strengthening*** atau **Penguatan MD.**

2.4.2.3 Inisialisasi MD-5

Pada MD-5 terdapat empat buah *word* 32 bit register yang berguna untuk menginisialisasi *message digest* pertama kali. Register-register ini diinisialisasikan dengan bilangan hexadesimal.

word A: 01 23 45 67
word B: 89 AB CD EF
word C: FE DC BA 98
word D: 76 54 32 10

Register-register ini biasa disebut dengan nama ***Chain variabel*** atau **variabel rantai**.

2.4.2.4 Proses Pesan di dalam Blok 16 Word

Pada MD-5 juga terdapat 4 (empat) buah fungsi nonlinear yang masing-masing digunakan pada tiap operasinya (satu fungsi untuk satu blok), yaitu:

$$F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$

$$H(X,Y,Z) = X \oplus Y \oplus Z$$

$$I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$$

($\oplus$ untuk XOR, $\wedge$ untuk AND, $\vee$ untuk OR dan $\neg$ untuk NOT).

Berikut dapat dilihat satu buah operasi dari MD-5 dengan operasi yang dipakai sebagai contoh adalah FF(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a + F(b,c,d) + Mj + ti) <<<s).

Bila Mj menggambarkan pesan ke-j dari sub blok (dari 0 sampai 15) dan <<<s menggambarkan bit

akan digeser ke kiri sebanyak s bit, maka keempat operasi dari masing-masing ronde adalah:

FF(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a + F(b,c,d) + Mj + ti) <<< s)

GG(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a + G(b,c,d) + Mj + ti) <<< s)

HH(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a + H(b,c,d) + Mj + ti) <<< s)

II(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a + I(b,c,d) + Mj + ti) <<< s)

Konstanta ti didapat dari integer 2^{32}. abs(sin(i)), dimana i dalam radian.

2.4.2.5 Keluaran MD-5

Keluaran dari MD-5 adalah 128 bit dari word terendah A dan tertinggi word D masing-masing 32 bit.

3. HASH-HASH MD5

Hash-hash MD5 sepanjang 128 bit (16 byte), yang dikenal juga sebagai ringkasan pesan, secara tipikal ditampilkan dalam bilangan heksadesimal 32 digit. Berikut ini merupakan contoh pesan ASCII sepanjang 43 byte sebagai masukan dan hash MD5 terkait :

MD5 (”The quick brown fox jump over the lazy dog”) = 9e107d9d372bb6826bd81d3542a419d6

Bahkan perubahan yang kecil pada pesan akan (dengan probabilitas lebih) menghasilkan hash yang benar-benar berbeda, misalnya pada kata ”dog”, huruf d diganti menjadi c :

MD5 (”The quick brown fox jump over the lazy cog”) = 1055d3e698d289f2af8663725127bd4b

Hash dari panjang nol adalah :

MD5(“”)=d41d8cd98f00b204e9800998ecf8427e

4. PENGUJIAN INTEGRITAS

MD5 digunakan secara luas dalam dunia perangkat lunak untuk menyediakan semacam jaminan bahwa file yang diambil (*download*) belum terdapat perubahan. Seorang *user* dapat membandingkan MD5 *sum* yang dipublikasikan dengan *checksum* dari file yang diambil. Dengan asumsi bahwa *checksum* yang dipublikasikan dapat dipercaya akan keasliannya, seorang *user* dapat secara yakin bahwa file tersebut adalah file yang sama dengan file yang dirilis oleh para *developer*, jaminan perlindungan dari *Trojan Horse* dan virus komputer yang ditambahkan pada perangkat lunak. Bagaimanapun juga, seringkali kasus yang terjadi bahwa *checksum* yang dipublikasikan tidak dapat dipercaya (sebagai contoh, *checksum* didapat dari channel atau lokasi yang sama dengan tempat mengambil file), dalam hal ini MD5 hanya mampu melakukan *error-checking*. MD5 akan mengenali file yang didownload tidak sempurna, cacat atau tidak lengkap.

5. KESIMPULAN

Kesimpulan yang dapat diperoleh dari makalah ini, yaitu :

1. *Message Digest 5* (MD5) adalah sebuah fungsi hash satu arah yang mengubah masukan dengan panjang variabel menjadi keluaran dengan panjang tetap yaitu 128 bit.
2. MD5 merupakan pengembangan lebih lanjut dari MD4 dimana terjadi penambahan satu ronde
3. Simplicity. Algoritma MD5 mudah untuk diimplementasikan karena tidak membutuhkan program yang besar dan panjang
4. Kecepatan enkripsi pada sistem kriptografi MD5 sangat bergantung kepada spesifikasi komputer yang digunakan
5. MD5 akan menghasilkan *output* berupa 4 buah blok yang masing-masing terdiri dari 32 bit sehingga menjadi 128 bit yang disebut nilai hash

DAFTAR PUSTAKA

[1] Message Digest 5. (2007).
http://en.wikipedia.org/wiki/MD5.
Tanggal Akses 26 Desember 2007
Pukul 21.00

[2] Hash Function. (2007).
http://en.wikipedia.org/wiki/Hash_function.
Tanggal Akses 26 Desember 2007
Pukul 21.00

[3] Cryptography. (2007).
http://en.wikipedia.org/wiki/Cryptography.
Tanggal Akses 26 Desember 2007
Pukul 21.00

[4] Munir, Rinaldi. 2004. Diktat Kuliah IF2153 Matematika Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.
Tanggal Akses 26 Desember 2007
Pukul 21.00

[5] Ilmu Komputer. (2007).
http://ilmukomputer.com/2007/03/14/md5-dan-sha-1-kriptografi-dengan-fungsi-hash/.
Tanggal Akses 26 Desember 2007
Pukul 21.00

[6] Elektro Undip. (2007).
http://www.elektro.undip.ac.id/transmisi/jun06/5_aghus_abp.pdf.
Tanggal Akses 26 Desember 2007
Pukul 21.00