# Analisis Cara Kerja Beragam Fungsi Hash Yang Ada

Christian Angga – 13508008

*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*ca_ers@hotmail.com*

***Abstract*— Dewasa ini, pada dasarnya fungsi hash sangat berguna dan banyak sekali dipakai dalam kehidupan sehari-hari. Fungsi hash adalah fungsi yang menerima masukan berupa string sepanjang apapun lalu mengkonversikannya menjadi string dengan panjang keluaran tetap. Banyak sekali kegunaan dari fungsi hash yang ada, diantaranya adalah untuk autentifikasi password, autentifikasi keaslian file, tanda tangan digital, dan sebagainya. Namun secara garis besar, fungsi hash ini diciptakan sebagai alat untuk autentifikasi atau pengecekan.**

**Fungsi hash merupakan sebuah fungsi yang bekerja satu arah (tidak dapat dikembalikan ke kondisi/string semula). Sudah ada cukup banyak fungsi hash yang ada di kehidupan kita ini, beberapa diantaranya yang cukup terkenal di dunia adalah MD2, MD4, MD5, RIPEMD, SHA-0, SHA-1, SHA-256, SHA-512, WHIRLPOOL. Selain itu, masih ada banyak lagi fungsi hash yang ada di dunia ini, bahkan pada dasarnya, setiap orang mampu untuk membuat fungsi hashnya sendiri. Dari semua algoritma fungsi hash yang ada ini, pada dasarnya ada kemiripan dan kesamaan cara kerja antara satu fungsi hash dan fungsi hash lainnya.**

**Pada makalah ini, penulis mencoba untuk menganalisis tiap-tiap fungsi komponen yang terdapat di dalam fungsi-fungsi hash pada umumnya. Tiap-tiap komponen tersebut pada dasarnya memiliki fungsi masing-masing yang pada akhirnya akan digabung menjadi satu agar fungsi hash final yang dihasilkan benar-benar teracak secara random, aman, tidak terjadi kolisi, dan tidak ada kriptanalis yang dapat membobolnya.**

***Index Terms*—Fungsi Hash, Komponen, Cara Kerja.**

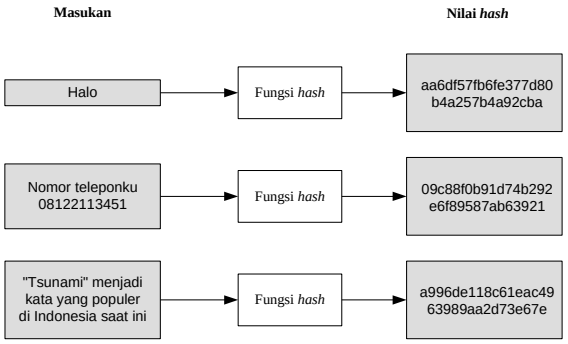
## I. INTRODUCTION

Fungsi hash adalah fungsi yang menerima masukkan string yang panjangnya sembarang dan mengkonversinya menjadi string keluaran yang panjangnya tetap atau fixed. Fungsi hash yang dihasilkan biasanya dituliskan dalam notasi persamaan sebagai berikut:

$$h = H(M)$$

Pada persamaan di atas, h merupakan nilai hash yang dihasilkan, sedangkan H adalah fungsi hash-nya itu sendiri, dan M adalah message atau pesan yang akan diubah dan dikonversikan menjadi nilai hash (hash value). Nilai hash yang dihasilkan biasa disebut juga pesan ringkas atau message digest. Fungsi hash dapat mengkonversikan sembarang pesan yang berukuran berapa saja menjadi message digest yang berukuran tetap, dan biasanya lebih pendek dari panjang pesan yang asli/semula.

Berikut contoh penggunaan fungsi hash yang mengubah suatu string dengan panjang berapapun menjadi sebuah message digest dengan panjang tetap:




**Gambar 1. Contoh Penggunaan Fungsi Hash**

Fungsi hash pada dasarnya bekerja satu arah, berarti pesan asli atau pesan semula akan diubah menjadi sebuah message digest namun message digest yang dihasilkan tidak dapat dikembalikan menjadi pesan asli atau pesan semula kembali. Fungsi hash satu arah ini pada dasarnya sudah banyak dibuat orang. Beberapa diantaranya yang cukup terkenal di dunia adalah MD2, MD4, MD5, RIPEMD, SHA-0, SHA-1, SHA-256, SHA-512, WHIRLPOOL. Namun beberapa diantaranya memiliki kelemahan yaitu kolisi dan dapat diserang oleh kriptanalis.

Fungsi hash pada dasarnya telah banyak digunakan di dalam kehidupan sehari-hari dalam berbagai aplikasi seperti autentifikasi password, autentifikasi keaslian file, tanda tangan digital, dan sebagainya. Kebanyakan fungsi hash ini digunakan sebagai autentifikasi, dan pada dasarnya autentifikasi ini sangat penting dan biasanya file-file yang diautentifikasi itu merupakan file penting atau rahasia, sehingga pada dasarnya, fungsi hash yang ada haruslah benar-benar memiliki tingkat keamanan yang tinggi.

## II. TEORI DASAR

Hash adalah suatu teknik "klasik" dalam Ilmu Komputer yang banyak digunakan dalam praktek secara mendalam. Hash merupakan suatu metode yang secara langsung mengakses record-record dalam suatu tabel dengan melakukan transformasi aritmatik pada key yang menjadi alamat dalam tabel tersebut. Key merupakan suatu input dari pemakai di mana pada umumnya berupa nilai atau string karakter.

Pelacakan dengan menggunakan Hash terdiri dari dua langkah utama, yaitu:

1. Menghitung Fungsi Hash
   Fungsi Hash adalah suatu fungsi yang mengubah key menjadi alamat dalam tabel. Fungsi Hash memetakan sebuah key ke suatu alamat dalam tabel. Idealnya, key-key yang berbeda seharusnya dipetakan ke alamat-alamat yang berbeda juga. Pada kenyataannya, tidak ada fungsi Hash yang sempurna. Kemungkinan besar yang terjadi adalah dua atau lebih key yang berbeda dipetakan ke alamat yang sama dalam tabel. Peristiwa ini disebut dengan collision (tabrakan). Karena itulah diperlukan langkah berikutnya, yaitu collision resolution (pemecahan tabrakan).

2. Collision Resolution
   Collision resolution merupakan proses untuk menangani kejadian dua atau lebih key di-hash ke alamat yang sama. Cara yang dilakukan jika terjadi collision adalah mencari lokasi yang kosong dalam tabel Hash secara terurut. Cara lainnya adalah dengan menggunakan fungsi Hash yang lain untuk mencari lokasi kosong tersebut.

Jenis fungsi hash:
Fungsi Hash (dilambangkan dengan h(key)) bertugas untuk mengubah key menjadi suatu nilai dalam interval [0....LEVELSIZE-1], dimana "LEVELSIZE" adalah jumlah maksimum dari record-record yang dapat ditampung dalam tabel. Jumlah maksimum ini bergantung pada ruang memori yang tersedia. Fungsi Hash yang ideal adalah mudah dihitung dan bersifat random, agar dapat menyebarkan semua key. Dengan key yang tersebar, berarti data dapat terdistribusi secara seragam dan collision dapat dicegah. Sehingga kompleksitas kinerja model Hash dapat mencapai O(1), di mana kompleksitas tersebut tidak ditemukan pada struktur model lain.

Penanggulangan Collision:
Masalah yang biasanya terjadi adalah, data yang ada begitu besar dibandingkan dengan jumlah lokasi dalam tabel. Sangatlah sukar, bahkan tidak mungkin untuk menemukan fungsi Hash yang dapat mencegah collision. Oleh karena itu penanggulangan collision yang baik sangat penting dalam Hash agar bisa meminimalkan jumlah collision.

Jika tabel Hash yang dimiliki berukuran T[0...LEVELSIZE-1], collision terjadi apabila lokasi T[h(key)] telah terisi pada saat ingin menyisipkan key. Oleh karena itu, harus ada suatu cara untuk menentukan lokasi lain dalam tabel Hash tersebut untuk menyimpan key. Collison resolution bertujuan untuk menentukan lokasi kosong tersebut.

Teknik-teknik collision yang ada diantaranya adalah:

1. Separate Chaining
   Teknik Separate Chaining merupakan suatu teknik untuk mengatasi collision dengan cara menggunakan daerah di luar tabel Hash dalam bentuk linier. Bentuk linier ini bisa berupa linked-list atau array. Urutan data pada daerah overflow bisa terurut atau random, tergantung cara penyisipan yang dilakukan.

2. Open Addressing
   Teknik Open Addressing adalah suatu teknik penyimpanan dalam tabel Hash yang membutuhkan ruang memori sangat besar. Open Addressing Hash menyimpan N data dalam tabel Hash yang berukuran LEVELSIZE dengan menggunakan tempat-tempat kosong untuk menangani collision resolution.

3. Double Hashing
   Teknik Double Hashing adalah suatu teknik penyimpanan dalam tabel Hash dimana jika terjadi collision, lokasi selanjutnya adalah lokasi sekarang + h(k). h(k) adalah suatu fungsi Hash yang berbeda dengan fungsi Hash yang pertama.

Kelebihan dan Kelemahan model hashing:
Model Hash memiliki kelemahan pada saat penambahan dan penghapusan record. Diperlukan waktu untuk menstrukturisasi tabel Hash. Pada tabel Hash terdapat suatu daerah (akibat collision) kelebihan data (overflow). Untuk mencapai daerah tersebut, memerlukan waktu. Lamanya waktu disebabkan jika data diletakkan dalam disk atau tape. Hal ini bisa diatasi dengan meletakkan keseluruhan data sampel secara utuh ke dalam memori. Namun ruang memori yang terbatas merupakan kendala tersendiri, karena data yang tersedia dalam kamus kata ini sangat besar (46.010 buah).

Penghapusan suatu unsur dari tabel Hash dapat menimbulkan masalah. Penghapusan suatu key dapat menyebabkan key-key yang berikutnya tidak dapat diakses, karena stuktur tabel yang sudah ada menjadi berubah. Namun dalam pembuatan kamus kata, hal tersebut tidak perlu dikhawatirkan, karena program ini tidak membutuhkan suatu penghapusan. Semua data yang terdapat dalam "Basis Data Kamus Kata" (yang berarti berada dalam tabel Hash), bersifat permanen, sehingga tidak perlu penghapusan.

## III. PROSES CARA KERJA ALGORITMA FUNGSI HASH

### A. Algoritma MD5

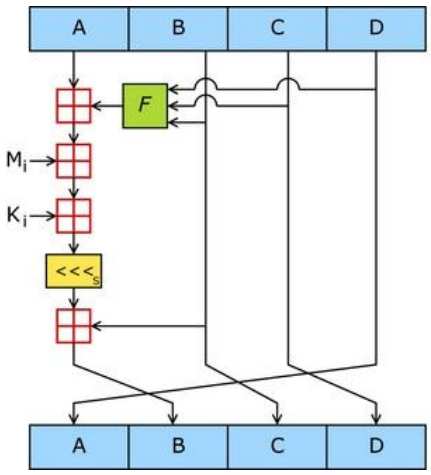
MD5 ialah fungsi hash kriptografik yang digunakan secara luas dengan hash value128-bit. Pada standard Internet (RFC 1321), MD5 telah dimanfaatkan secara bermacam-macam pada aplikasi keamanan, dan MD5 juga umum digunkan untuk melakukan pengujian integritas sebuah file.

MD5 di desain oleh Ronald Rivest pada tahun 1991 untuk menggantikan hash function sebelumnya, yaitu MD4 yang berhasil diserang oleh kriptanalis. Algoritma MD5 menerima masukan berupa pesan dengan ukuran sembarang dan menghasilkan message digest yang panjangnya 128 bit.

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 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 dan didefinisikan pada RFC 1321. 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. 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.

Langkah-langkah pembuatan message digest dengan algoritma MD5 adalah sebagai berikut:




**Gambar 2. Cara Kerja MD5**

1. Penambahan Bit-bit Pengganjal
   a. Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512.
   b. Jika panjang pesan 448 bit, maka pesan tersebut ditambah dengan 512 bit menjadi 960 bit. Jadi, panjang bit-bit pengganjal adalah antara 1 sampai 512.
   c. Bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

2. Penambahan Nilai Panjang Pesan Semula
   a. Pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula.
   b. Jika panjang pesan > 264 maka yang diambil adalah panjangnya dalam modulo 264. Dengan kata lain, jika panjang pesan semula adalah K bit, maka 64 bit yang ditambahkan menyatakan K modulo 264.
   c. Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512 bit.

3. Inisialisai Penyangga MD
   MD5 membutuhkan 4 buah penyangga (buffer) yang masing-masing panjangnya 32 bit. Total panjangpenyangga adalah 4 ´ 32 = 128 bit. Keempat penyangga ini menampung hasil antara dan hasil akhir.

   Keempat penyangga ini diberi nama A, B, C, dan D. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

   A = 01234567
   B = 89ABCDEF
   C = FEDCBA98
   D = 76543210

4. Pengolahan Pesan dalam Blok Berukuran 512 bit.
   Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y0 sampai YL – 1).
   Setiap blok 512-bit diproses bersama dengan penyangga MD menjadi keluaran 128-bit, dan ini disebut proses HMD5.

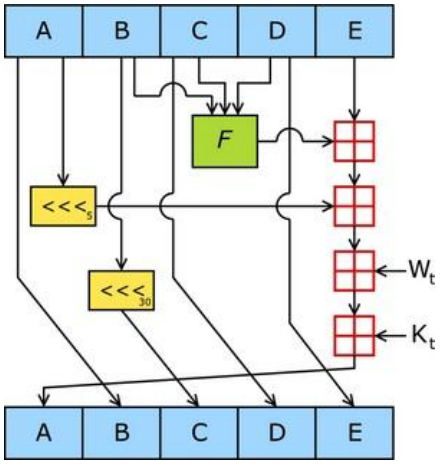
B. Algoritma SHA-512

Algoritma SHA-512 adalah algoritma yang menggunakan fungsi hash satu arah yang diciptakan oleh Ron Rivest. Algoritma merupakan pengembangan dari algoritma-algoritma sebelumnya yaitu algoritma SHA-0, SHA-1, SHA-256 dan algoritma SHA-384.

Beberapa contoh algoritma fungsi hash yang umum digunakan antara lain MD4, SHA1 dan MD5 yang merupakan perbaruan dari MD4. Keterkaitan dan perkembangan dari algoritma hash tersebut, menunjukkan bahwa algoritma tersebut terbukti telah terjadi collision. Saat ini, National Institute of Standard and Technology (NIST) telah menjadikan SHA-224, SHA-256, SHA-384, dan SHA-512 sebagai standard fungsi hash yang baru. Standard fungsi hash yang baru saat ini masih terus dalam kajian apakah fungsi hash ini bersifat collision free karena fungsi tersebut masih turunan dari MD5 yang telah terbukti bersifat collision.

Cara kerja kriptografi algoritma SHA-512 adalah menerima input berupa pesan dengan ukuran sembarang dan menghasilkan message diggest yang memiliki panjang 512 bit. Berikut illustrasi gambar dari pembuatan message diggest pada kriptografi algoritma SHA-512 :

Langkah-langkah pembuatan message digest dengan algoritma SHA-512 adalah sebagai berikut:




**Gambar 3. Cara Kerja SHA-512**

1. Penambahan Bit-bit Pengganjal
   Proses pertama yang dilakukan adalah menambahkan pesan dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 896 mod 1024 Ini berarti setelah menambahkan bit-bit pengganjal, kini panjang pesan adalah 128 bit kurang dari kelipatan 1024. Hal yang perlu diingat adalah angka 1024 muncul karena algoritma SHA-512 memproses pesan dalam blok-blok yang berukuran 1024.

   Apabila terdapat pesan dengan panjang 24 bit, maka pesan tersebut akan tetap ditambahkan dengan bit-bit pengganjal. Pesan akan ditambahkan dengan 896 - (24 + 1) = 871 bit. Jadi panjang bit-bit pengganjal adalah antara 1 sampai 896. Lalu satu hal lagi yang perlu diperhatikan adalah bahwasanya bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

2. Penambahan Nilai Panjang Pesan Semula
   Kemudian proses berikutnya adalah pesan ditambah lagi dengan 128 bit yang menyatakan panjang pesan semula. Apabila panjang pesan lebih besar dari 2128 maka yang diambil adalah panjangnya dalam modulo 2128. Dengan kata lain, jika pada awalnya panjang pesan sama dengan K bit, maka 128 bit yang ditambahkan menyatakan K modulo 2128. sehingga setelah proses kedua ini selesai dilakukan maka panjang pesan sekarang adalah 1024 bit.

3. Inisialisasi Nilai Hash
   Pada algoritma SHA-512 nilai Hash, H(0) terdiri dari 8 words dengan besar 64 bit dalam notasi hexadesimal sebagai berikut :

| Penyangga | Nilai Awal () |
|---|---|
| A | 6a09e667f3bcc908 |
| B | bb67ae8584caa73b |
| C | 3c6ef372fe94f82b |
| D | a54ff53a5f1d36f1 |
| E | 510e527fade682d1 |
| F | 9b05688c2b3e6c1f |
| G | 1f83d9abfb41bd6b |
| H | 5be0cd19137e2179 |

## IV. ANALISIS DAN PERBANDINGAN

### A. Penambahan Bit-bit Pengganjal

Pada dasarnya, dari kedua algoritma fungsi hash di atas (MD5 dan SHA-512), keduanya memiliki komponen utama yang sama, yaitu penambahan bit-bit pengganjal. Pada algoritma MD5, penambahan bit pengganjal ini ditambakan sedemikian sehingga panjang pesan kongruen dengan 448 modulo 512. Sedangkan pada algoritma SHA-512, penambahan bit pengganjal ditambahkan sedemikian sehingga panjang pesan kongruen dengan 896 mod 1024. Dapat dilihat disini bahwa algoritma SHA-512 menggunakan panjang bit tepat dua kali lipat dengan algoritma MD5.

Jika dilihat dari komposisi bit-bit pengganjal yang dimiliki oleh algoritma MD5 dan SHA-512, keduanya memiliki kesamaan komposisi, yaitu terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

Dari kedua algoritma fungsi hash ini, pada dasarnya keduanya memiliki komponen utama yang sama, yaitu penambahan bit-bit pengganjal. Mengapa komponen ini diperlukan? Komponen ini diperlukan karena algoritma MD5 memproses pesan dalam blok-blok yang berukuran 512 bit, sedangkan algoritma SHA-512 memproses pesan dalam blok-blok yang berukuran 1024.

### B. Penambahan Nilai Panjang Pesan Semula

Dari kedua algoritma fungsi hash di atas (MD5 dan SHA-512), keduanya memiliki komponen kedua yang sama, yaitu penambahan nilai panjang pesan semula. Pada algoritma MD5, pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Sedangkan pada algoritma SHA-512, pesan ditambah lagi dengan 128 bit yang menyatakan panjang pesan semula. Dapat dilihat dengan jelas, terdapat kesamaan komponen dari kedua algoritma ini, namun pada algoritma MD5, hanya ditambah lagi dengan 64 bit, sedangkan pada algoritmat SHA-512, ditambahkan lagi dengan 128 bit yang merupakan tepat dua kali lipat dari penambahan yang dilakukan oleh algoritma MD5.

Jikalau panjang pesan melebihi batas yang dimiliki oleh masing-masing algoritma di atas, maka akan diambil sepanjang modulo yang diperlukan oleh masing-masing algoritma tersebut. Pada algoritma MD5, jika panjang pesan > 264 maka yang diambil adalah panjangnya dalam modulo 264. Dengan kata lain, jika panjang pesan semula adalah K bit, maka 64 bit yang ditambahkan menyatakan K modulo 264. Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512 bit. Sedangkan pada algoritma SHA-512, apabila panjang pesan lebih besar dari 2128 maka yang diambil adalah panjangnya dalam modulo 2128. Dengan kata lain, jika pada awalnya panjang pesan sama dengan K bit, maka 128 bit yang ditambahkan menyatakan K modulo 2128. sehingga setelah proses kedua ini selesai dilakukan maka panjang pesan sekarang adalah 1024 bit. Tampak jelas persamaan konsep dari kedua algoritma tersebut, namun diantara keduanya, hanya memiliki perbedaan kelipatan bit pada akhirnya. Pada algoritma MD5 panjang pesan sekarang adalah 512 bit. Sedangkan pada SHA-512, panjang pesan sekarang adalah 1024 bit, tepat dua kali lipat dari panjang pesan yang dimiliki oleh MD5.

### C. Inisialisasi

Dari kedua algoritma fungsi hash di atas (MD5 dan SHA-512), keduanya memiliki komponen ketiga yang sama juga, yaitu inisialisasi. Pada MD5, MD5 membutuhkan 4 buah penyangga (buffer) yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 4 * 32 = 128 bit. Keempat penyangga ini menampung hasil antara dan hasil akhir. Keempat penyangga ini diberi nama A, B, C, dan D. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

A = 01234567
B = 89ABCDEF
C = FEDCBA98
D = 76543210

Tidak berbeda jauh juga antara MD5 dan SHA-512, inisialisasi nilai hash pada algoritma SHA-512 terdiri dari 8 words dengan besar 64 bit dalam notasi hexadesimal sebagai berikut :

| Penyangga | Nilai Awal () |
|---|---|
| A | 6a09e667f3bcc908 |
| B | bb67ae8584caa73b |
| C | 3c6ef372fe94f82b |
| D | a54ff53a5f1d36f1 |
| E | 510e527fade682d1 |
| F | 9b05688c2b3e6c1f |
| G | 1f83d9abfb41bd6b |
| H | 5be0cd19137e2179 |

Dari kedua algoritma fungsi hash di atas (MD5 dan SHA-512), penyangga yang dimiliki SHA-512 pun ternyata tepat dua kali dari penyangga yang dimiliki oleh MD5. Baik dari panjangnya, maupun dari jumlahnya.

## V. CONCLUSION

Dilihat dari cara kerja dan komponen-komponen utama yang dimiliki oleh kedua algoritma tersebut (MD5 dan SHA-512), dapat disimpulkan bahwa SHA-512 adalah perpanjangan dari algoritma MD5.

Kegunaan dari masing-masing komponen tersebut pada dasarnya mengacu pada keamanan. Dengan mengembangkan MD5 menjadi SHA-512 dengan konsep yang sama namun dikembangkan dengan pendekatan SHA-512 dua kali lipat MD5, hal ini mengakibatkan keamanan serangan terhadap brute-force meningkat. Dengan demikian, tingkat keamanan yang dimiliki jauh meningkat.

Kedua algoritma tersebut simple untuk dijelaskan dan mudah untuk diiemplementasikan karena tidak membutuhkan program yang besar atau tabel subtitusi yang besar pula, namun memiliki tingkat keamanan yang tinggi.

## REFERENCES


Fungsi Hash
<http://www.scribd.com/doc/29412263/Fungsi-Hash>
Waktu akses : 26 April 2011.

Cara Kerja MD5 Hash
< http://www.2lisan.com/read/cara-kerja-md5-hash>
Waktu akses : 26 April 2011.

Algoritma SHA
<http://www.forumsains.com/ilmu-komputer/algoritma-sha-1/?wap2>
Waktu akses : 26 April 2011.

Perbedaan SHA1 dan MD5
<http://dark-holi.blogspot.com/2011/03/perbedaan-antara-md5-dan-sha.html>
Waktu akses : 26 April 2011.

SHA1 Overview
< http://stsn6.wordpress.com/2009/10/15/sha-1-overview/>
Waktu akses : 26 April 2011.

FUNGSI HASH PADA KRIPTOGRAFI
<http://egiewendra.blog.upi.edu/files/2009/06/fungsi-hash-pada-kriptografi.pdf>
Waktu akses : 26 April 2011.


## 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, 9 Mei 2011

Christian Angga
13508008