APLIKASI KRIPTOGRAFI DENGAN ALGORITMA MESSAGE DIGEST 5(MD5)

YONGKE YOSWARA
13508034
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
if18034@students.if.itb.ac.id

ABSTRAK

Pengiriman pesan sekarang ini banyak menggunakan jaringan untuk memudahkan kita. Masalah keamanan dan kerahasiaan dari suatu pesan merupakan aspek yang paling penting dari proses pengiriman informasi. Ketika menerima atau mengirim pesan, terdapat beberapa persoalan yang sangat penting, yaitu: kerahasiaan(*confidentiality*), integritas data(menjamin pesan tidak diubah oleh orang lain), keaslian pesan (*aunthentication*), dan tak terbantahkan (*non-repudiation*). Pesan menjadi tidak berguna apabila pesan tersebut dibajak atau disadap oleh orang lain di tengah jalan. Oleh sebab itu, diperlukan suatu teknik kriptografi untuk menangani persoalan ini. Salah satu algoritma yang dipakai adalah Message Digest 5 (MD5). MD5 merupakan pengembangan dari MD4. MD5 adalah fungsi hash kriptografi yang banyak digunakan sebagai alat untuk menjamin dan memberi garansi bahwa pesan yang dikirim akan sesuai dengan pesan yang diterima dengan cara membandingkan "sidik jari" dari kedua pesan tersebut. MD5 memproses teks masukan ke dalam blok-blok bit sebanyak 512 bit, yang kemudian akan 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 (biasa disebut nilai hash). Sedangkan kriptografi adalah ilmu untuk menjaga kerahasiaan suatu berita. Istilah yang sering digunakan dalam kriptografi adalah plainteks, chiperteks, enkripsi, dan deskripsi.

Kata kunci: Message Digest 5, kriptografi, "sidik jari"

1. PENDAHULUAN

Keamanan dan kerahasiaan data pada jaringan komputer saat ini menjadi isu yang sangat penting. Ada banyak kasus menyangkut dengan keamanan dan kerahasiaan jaringan komputer. Banyak orang dibayar untuk menjadi *hacker*, dimana mereka bisa masuk ke dalam data pada jaringan komputer saat ini. Selain itu, cukup besar dana yang dibutuhkan untuk membiayai pengamanan data pada jaringan komputer supaya tidak mudah "ditembus" oleh para *hacker*. Sistem pertahanan, sistem perbankan, dan sistem lainnya membutuhkan tingkat keamanan dan penanganan yang tinggi. Kemajuan bidang jaringan komputer dianggap sebagai salah satu pemicu semakin maraknya sistem komputer yang "dirusak" oleh *hacker*.

Untuk menjaga keamanan dan kerahasian dari pesan atau data, ataupun informasi dalam jaringan komputer, maka kita memerlukan suatu teknik penyandian supaya pesan tersebut tidak dapat dimengerti oleh orang lain. Teknik penyandian itu adalah enkripsi dan deskripsi. Enkripsi adalah mengubah data asli (plainteks) menjadi data rahasia (chiperteks). Sedangkan deskripsi dilakukan saat penerimaan dengan mengubah data rahasia (chiperteks) menjadi data sebenarnya (plainteks). Hal ini menyebabkan pada saat pengiriman pesan, maka chiperteks-lah yang dikirim dengan menggunakan kunci rahasia.

Kriptografi merupakan salah satu cara untuk melakukan proses enkripsi dan deskripsi. Sejarah kriptografi telah ada sejak jama sebelum internet muncul. Yang paling menonjol adalah Kode Telegram Zimmermann. Kode ini berisi rentetan angka yang dikirim pemerintah Jerman ke pemerintah Meksiko. Isi dari telegram tersebut adalah ajakan kepada Meksiko untuk ikut serta "meramaikan" perang dengan melawan Amerika. Tetapi kode ini akhirnya dipecahkan oleh intelijen Inggris dan mengakibatkan Amerika ikut serta dalam Perang Dunia Pertama (sebelumnya Amerika bertindak netral).

Salah satu bagian dari kriptografi adalah fungsi hash satu arah (*one-way hash function*). Fungsi hash satu arah memudahkan kita dalam melakukan enkripsi untuk mendapatkan chiperteksnya tetapi akan sangat sulit untuk mendapatkan plainteksnya. Fungsi hash satu arah digunakan untuk membuat "sidik jari" dari suatu dokumen atau pesan X. Pesan X yang akan di-hash disebut dengan *pre-image*, sedangkan outputnya disebut dengan nilai hash dan mempunyai ukuran tetap dengan aslinya.

Salah satu fungsi hash yang paling banyak digunakan adalah Message Digest 5 (MD5). MD5 merupakan fungsi hash satu arah yang diciptakan oleh Ron Rivest pada tahun 1991 untuk menggantikan *hash function*

sebelumnya, yaitu MD4. Selain MD4 dan MD5, masih ada juga contoh lainnya, yaitu Message Digest 2. MD5 merupakan salah satu aplikasi yang digunakan untuk mengetahui bahwa pesan yang dikirim tidak ada perubahan sewaktu berada di jaringan.

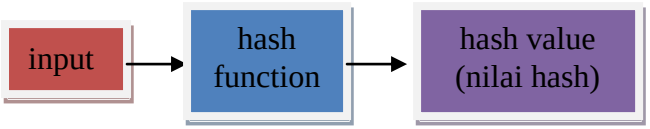
Algoritma MD5 secara garis besar adalah mengambil pesan yang mempunyai panjang variabel dan diubah menjadi "sidik jari" atau "intisari pesan" yang memiliki panjang tetap, yaitu 128 bit. "sidik jari" ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari "sidik jari" MD5 tersebut.

MD5 seringkali dipakai untuk enkripsi *password*, namun karena sudah banyak beredar tabel-tabel heksadesimal untuk membalikkan kode dari MD5, keamanan MD5 sebagai metode enkripsi *password* sudah sangat tidak aman.

Message Digest atau inti sari dari pesan harus mengandung tiga sifat penting, yaitu:

- ✓ Bila P diketahui, maka MD(P) akan dengan mudah 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) tidak sama dengan H(M')

Apabila diimplementasikan ke dalam bentuk diagram, maka fungsi hash satu arah akan tampah seperti diagram berikut ini:



Gambar 1. Diagram Fungsi Hash

2. Kriptografi dan Fungsi Hash Satu Arah

Pada bagian ini, akan dijelaskan secara lebih terperinci tentang bagian-bagian dari kriptografi, seperti enkripsi, deskripsi, chiperteks, plainteks, penggunaan kunci simetrik dan kunci asimetrik, prinsip dasar dan tujuan dari kriptografi, fungsi hash satu arah, dan juga algoritma Message Digest 5 itu sendiri.

2.1 Kriptografi

Kriptografi merupakan ilmu untuk menjaga kerahasiaan pesan (data atau informasi) dengan cara menyamarkannya menjadi bentuk tersandi yang tida mempunyai makna. Kriptografi digunakan untuk menyembunyikan pesan rahasia dari pihak yang dirahasiakan. Kriptografi mencegah "pembajakan" pesan oleh orang yang seharusnya tidak berhak untuk mengetahuinya. Kriptografi sudah digunakan sejak permulaan tahun 400 SM oleh tentara Sparta yang sedang berada di Yunani. Mereka menggunakan *scytale* (alat yang terdiri dari sebuah pita panjang dari daun *papyrus* yang dililitkan pada sebatang silinder).

Ada 4 tujuan sistem kriptografi, yaitu:

- *Confidentiality*
  Memberikan kerahasiaan pesan dan menyimpan data dengan menyembunyikan informasi melalui teknik-teknik enkripsi
- *Message Integrity*
  Memberikan jaminan untuk tiap bagian bahwa pesan tidak akan mengalami perubahan dari saat ia dibuat hingga saat ia dibuka
- *Non-repudiation*
  Memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang apabila ia mencoba menyangkal memiliki dokumen tersebut
- *Authentication*
  Memberikan dua layanan. Pertama, mengidentifikasi keaslian suatu pesan dan memberikan jaminan keautentikannya. Kedua, menguji identitas seseorang apabila ia akan memasuki sebuah sistem

Pada dasarnya keamanan dan kerahasian suatu pesan, data, ataupun informasi adalah merupakan hal mutlak yang harus dilakukan. Alat untuk melakukan pengamanan data dalam sistem komunikasi disebut dengan *cryptography*. Kriptografi merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Kriptanalisis (*cryptanalysis*) merupakan ilmu dan seni pembongkaran pesan, data, atau informasi rahasia. Sedangkan kriptologi (*cryptology*) adalah panduan dari kriptografi dari kriptografi dan kriptanalist.

Ada tiga buah komponen dalam kriptografi, yaitu:

1. *Plaintext* / plainteks : sumber berita/pesan
2. *Ciphertext*/ cipherteks : pesan yang telah diproses
3. Algoritma dan kunci

Ada dua buah bagian penting dalam kriptografi, yaitu:

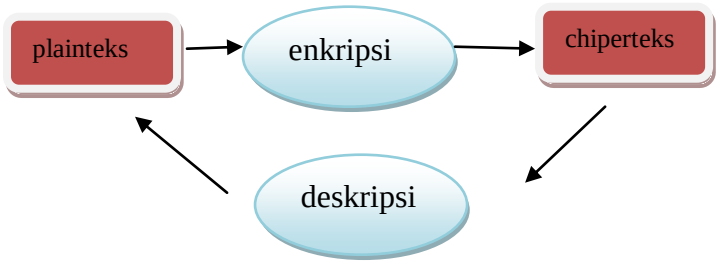
1. Enkripsi
2. Deskripsi

Enkripsi adalah proses dari penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti pesan aslinya. Sedangkan deskripsi adalah proses untuk mengembalikan pesan yang telah dienkripsi menjadi pesan yang aslinya. Jadi enkripsi dan deskripsi berlawanan. Dasar matematis yang mendasari proses enkripsi dan deksripsi adalah relasi antara dua buah himpunan yang berisi elemen teks asli (plainteks) dan yang berisi elemen sandi (chiperteks). Enkripsi dan deskripsi merupakan fungsi transformasi antara kedua himpunan tersebut. Sebagai contoh, kita menotasikan elemen teks asli sebagai A, elemen teks sandi sebagai B, dan proses enkripsi dengan E, kemudian proses deskripsi dengan D.

Enkripsi: E(A) = B
Deskripsi: D(B) = A

Yang dimasuk dengan plainteks adalah teks jelas yang dapat dimengerti. Teks tersebut merupakan pesan yang akan dirahasiakan. Chiperteks berarti teks yang sudah

disandikan. Berikut ini adalah gambar yang menunjukkan proses dari kriptografi secara umum:



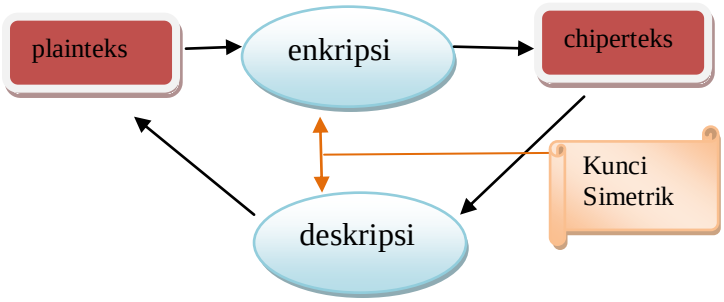
Gambar 2. Gambar Proses Enkripsi dan Deskripsi Kriptografi

Ada dua jenis kunci pada proses pengelolaan kriptografi, yaitu kunci simetrik dan kunci asimetrik.

2.1.1 Kunci Simetrik

Enkripsi yang menggunakan kunci simetrik sering disebut dengan enkripsi konvensional. Enkripsi ini menggunakan kunci yang sama untuk proses enkripsi dan deskripsi.

Penggunaan metode ini membutuhkan persetujuan antara pengirim dan penerima pesan tentang kunci yang akan digunakan selama proses enkripsi dan deskripsi. Persetujuan itu biasanya dilakukan sebelum mereka saling mengirim pesan. Apabila seorang penyusup dapat menemukan kunci simetrik dari proses kriptografi, maka prang tersebut dapat dengan mudah membaca chiperteks (pesan yang telah dienkripsi).

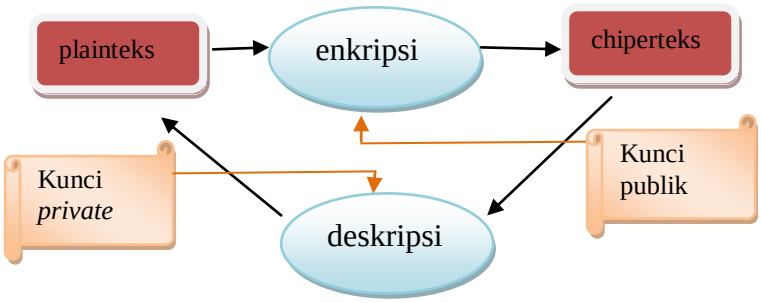


Gambar 3. Gambar Penggunaan Kunci Simetrik

2.1.2 Kunci Asimetrik

Enkripsi kunci asimetrik dibuat sedemikian rupa sehingga kunci yang dipakai untuk enkripsi dan deskripsi berbeda. Enkripsi kunci publik adalah kunci untuk enkripsi yang boleh disebarluaskan kepada umum. Sedangkan kunci untuk mendeskripsi hanya disimpan oleh orang yang bersangkutan.

Bila seseorang ingin mengirimkan pesan kepada orang lain, maka pengirim pesan akan menggunakan kunci publik untuk melakukan enkripsi. Sedangkan penerima pesan akan menggunakan kunci *private* miliknya untuk mendeskripsi pesan tersebut.



Gambar 4. Gambar Penggunaan Kunci Asimetrik

Cara membuat kunci publik dan kunci privat sederhana adalah sebagai berikut:

Pilihlah dua buah bilangan prima yang cukup besar (minimal 100 digit) secara acak. Kita notasikan kedua bilangan prima itu sebagai p dan q. Hitung n = pq. Bilangan n ini disebut dengan parameter sekuriti. Pilih sebuah bilangan k secara acak, tetapi bilangan k tersebut tidak boleh memiliki faktor pembagi yang sama (selain bilangan 1) dengan (p-1) (q-1). Bilangan k ini kita jadikan sebagai kunci privat kita. Hitunglah h sedemikian sehingga kh mod (p-1)(q-1) =1. Kita bisa menggunakan algoritma Euclid untuk menghitung h dengan sangat efisien. Bilangan n dan h kita sebar ke publik. h merupakan kunci publik. Sementara itu bilangan p dan q boleh dibuang asalkan jangan tersebar ke publik.

Sebuah sistem kriptografi yang baik harus memiliki karakteristik sebagai berikut: keamanan sistem terletak pada kerahasiaan kunci dan bukan pada algoritma yang digunakan, memiliki ruang kunci yang besar, menghasilkan chiperteks yang terlihat acak serta mampu menahan seluruh gangguan yang telah dikenal sebelumnya.

2.2 Fungsi Hash Satu Arah

Fungsi hash satu arah adalah fungsi hash yang memiliki beberapa sifat keamanan tambahan sehingga dapat dipakain untuk tujuan keamanan data. Fungsi hash adalah fungsi yang mengubah *string* input dengan panjang berhingga menjadi *string* output dengan panjang yang tetap. Output dari fungsi hash disebut dengan nilai hash (*hash value*).

Berikut ini ada 2 tahap dalam fungsi hash kriptografi:

a. Tahap *pre-image*: apabila diketahui nilai hash H maka sulit untuk mendapatkan M dimana H = hash(M)

b. Tahap *pre-image* kedua : apabila diketahui input m1, maka sulit untuk mencari input m2 (input m2 tidak sama dengan input m1) yang menyebabkan hash(m1) = hash (m2)

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

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

dimana 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).

Ada beberapa nama untuk fungsi hash satu arah, antara lain adalah fungsi penyusutan, intisari pesan, sidik jari, dan fungsi pembanding. Fungsi hash satu arah adalah sebuah fungsi hash yang berjalan hanya satu arah. Kita akan dengan mudah untuk menghitung nilai hash dari pre-image, tetapi akan sangat sulit bagi kita untuk membangkitkan pre-image dari nilai hash-nya.

Metode fungsi hash satu arah adalah berfungsi untuk 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.

Kita ambil contoh, ada pengirim (X) dan penerima pesan (Y). X akan mengirimkan pesan kepada Y. sebelum mengirim pesan rahasia tersebut, X melakukan hash pada pesannya untuk mendapatkan nilai hash. Kemudian dia mengirimkan pesan itu beserta nilai hash-nya. Kemudian Y melakukan hash untuk mencari nilai dari pesaan tersebut. Apabila terjadi perbedaan maka dapat disimpulkan bahwa sewaktu pengiriman telah terjadi perubahan pada pesan tersebut.

2.3 Message Digest 5

Message Digest 5 (MD5) melanjutkan seri sebelumnya, yaiut Message Digest 4 (MD4). MD4 memiliki pengaruh yang besar dalam seri MD5. MD5 memiliki panjang 128 bit yang ditulis tahun 1991 dan menjadi standar internet sampai 2004.

MD5 adalah salah satu penggunaan fungsi hash satu arah yang paling banyak digunakan. MD5 merupakan fungsi hash satu arah kelima yang dirancang oleh Ron Rivest. MD5 memproses teks masukan ke dalam blok-blok bit sebanyak 512 bit, kemudian dibagi ke dalam 32 bit sub blok sebanyak 16 buah. Output dari MD5 berupa 4 buah blok yang masing-masing berisi 32 bit dan akan menjadi 128 bit yang biasa disebut dengan nilai hash.

2.3.1 Aplikasi MD-5

MD5 digunakan secara luas di dunia perangkat lunak sebagai alat jaminan kebenaran *file* yang telah diunduh. Kebanyak *server* penyedia file selalu memberikan sebuah fungsi *checksum* yang telah dihitung dengan format MD5 untuk memeriksa kesesuaian *file*. Ketika pengunduh selesai mengunduh *file* yang diinginkan, maka kita hanya perlu memeriksa dengan *checksum* yang telah disediakan.

MD5 juga seringkali digunakan untuk enkripsi *password*. Tetapi keamanan penggunaan MD5 sebagai metode enkripsi *password* sudah sangat tidak aman karena sudah cukup banyak beredar tabel-tabel heksadesimal untuk membalik kode dari MD5.

Kebanyakan dari praktisi kriptografer menggunakan sedikit ‘garam’ dalam password MD5 mereka dan menggunakan hash-ing lebih dari sekali.

CRAM MD5, *challenge response authentication mechanism*, adalah sebuah mekanisme pengesahan *client* dengan MD5. Dalam prosesnya, CRAM MD5 mengirimkan *string* ke *client*, kemudian *client* membalas dengan mengirimkan *username* diikuti spasi dan rentetan kode 16 byte dengan notasi heksadesimal. Kode ini adalah *password* yang nantinya akan di autentikasi oleh *server*.

2.3.2 Algoritma MD5

5 langkah untuk menghitung intisari pesan adalah:

i. Menambahkan bit
ii. Penambahan panjang pesan
iii. Inisialisasi MD5
iv. Proses pesan dalam blok 16 word
v. Keluaran MD5

Pada dasarnya, MD5 menghasilkan kode dengan panjang tetap dari sebuah pesan sembarang dengan panjang sembarang pesan yang masuk dipecah menjadi fragmen-fragmen dengan panjang masing-masing 512 bit. Pesan tersebut haruslah dimanipulasi sehinggal dapat dibagi 512. Proses manipulasi ini menambahkan bit 1 ke akhir pesan, lalu menambahkan 0 sampai panjang pesan sama dengan kelipatan dari 512 dikurangi 64. 64 bit terakhir ini digunakan untuk menyimpan *integer* dari panjang pesan aslinya.

Algoritma utama dari MD5 bekerja dalam keadaan 128 bit, yang terbagi menjadi 32 bit word A, B, C, dan D. Kemudian algoritma akan bekerja di setiap fragmen 512 bit. Proses dalam setiap fragmen ini terdiri dari empat putaran. Setiap putaran terdiri dari 16 operasi berdasarkan fungsi F, penambahan modular dan rotasi bit kiri.

2.3.2.1 Menambahkan Bit

Panjang bit harus kongruen dengan 448 mod 512 oleh sebab itu kita harus menambahkan bit-bit tambahan pada pesan. Penambahan bit selalu dilakukan meskipun panjang dari pesan sudah kongruen dengan 448 mod 512 bit. Penambahan bit dilakukan dengan menambahkan ‘1’ di awal dan diikuti penambahan ‘0’ sebanyak yang diperlukan sehingga panjang pesan kongruen dengan 448 mod 512.

2.3.2.2 Menambahkan Panjang Pesan

Setelah penambahan bit, pesan juga membutuhkan penambahan 64 bit supaya kongruen dengan kelipatan 512 bit. Bit-bit ini ditambahkan ke dalam dua *word* (masing-masing 32 bit) dan ditambahkan dengan *low-order*

terlebih dahulu. Penambahan pesan ini biasa disebut dengan *MD strengthening* atau penguatan MD.

2.3.2.3 Inisialisasi MD5

Pada MD5 terdapat empat buah *word* (masing-masing 32 bit) yang berguna untuk menginisialisasi MD5 pertama kali. Inisialisasi ini dilakukan dengan menggunakan bilangan heksadesimal.

- *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 variable* atau variabel rantai

2.3.2.4 Proses Pesan di Dalam Blok 16 Word

Pada MD5 terdapat 4 buah fungsi *non-linear* yang masing-masing digunakan pada tiap operasinya (satu jenis fungsi untuk satu blok), yaitu:

- F(X,Y,Z) = (X ∧ Y) ∨ ((¬ X) ∧ Z)
- G(X,Y,Z) = (X ∧ Z) ∨ (Y ∧ (¬ Z))
- H(X,Y,Z) = X ⊕ Y ⊕ Z
- I (X,Y,Z) = Y ⊕ (X ∨ (¬ Z))

(⊕ untuk XOR, ∧ untuk AND, ∨ untuk OR dan ¬ untuk NOT).

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 sebagai berikut:

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

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

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

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

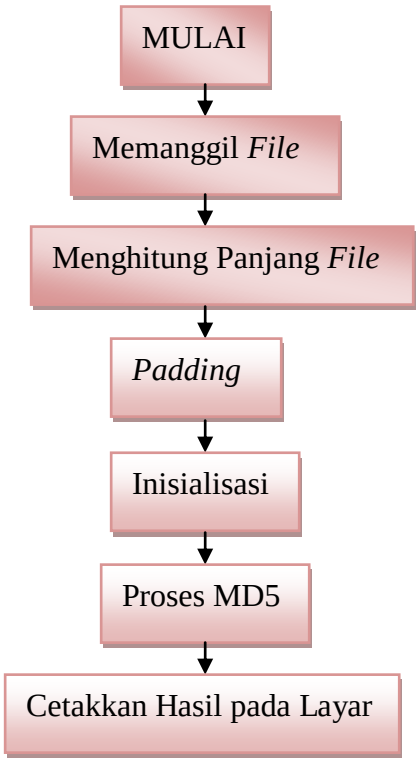
2.3.2.5 Keluaran MD5

Keluaran dari MD5 adalah 128 bit dari *word* terendah A dan tertinggi *word* D yang masing-masing 32 bit.

3 Proses MD5

3.1 Proses MD5 Sebagai Masukan Berupa File

Ada beberapa contoh masukan, bisa berupa teks, *file*, ataupun *test suite*. Tetapi kita ambil contoh apabila masukan yang menjadi *input* untuk proses MD5 adalah masukan berupa *file*. Proses ini memanggil *file* yang kemudian dihitung berapa panjang bitnya, *file* ini dianggap sebagai bit memori sehingga masukannya tidak dipengaruhi oleh eksistensinya. Kemudian dilakukan proses MD5. Penjelasan melalui gambar dapat dilihat di bawah ini.



Gambar 5. Proses MD5 dengan Masukkan Berupa File

3.2 Pengukuran Kecepatan Aplikasi

Mengukur kecepatan aplikasi merupakan sebuah analisa yang akan dipakai untuk mengukur tingkat kecepatan dari proses mencari nilai hash dari sebuah pesan/ *file* dengan menggunakan aplikasi MD5.

Rumus yang dipakai dalam aplikasi untuk menghitung kecepatan mencari nilai hash adalah:

- Kecepatan=Besar Ukuran File/Lama Waktu Proses

Satuan dari kecepatan enkripsi ini adalah Mbytes/detik. Dalam analisa kecepatan ini, akan dilakukan sebanyak lima kali pengambilan waktu terbaik yang diperlukan untuk enkripsi dalam setiap file dan kemudian kita mencari wkatu rata-ratanya.

3.3 Pengujian 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 dari hash MD5:

MD5(“The quick brown fox jumps over the lazy dog”). Nilai hash-nya : 9e107d9d372bb6826bd81d3542a419d6

Perubahan yang sangat kecil pasa satu buat input akan mengubah nilai hash secara keseluruhan. Kita ambil contoh huruf ‘d’ pada kata ‘dog’ kita ubah menjadi huruf ‘c’, maka yang terjadi adalah sebagai berikut:

MD5("The quick brown fox jumps over the lazy cog") = 1055d3e698d289f2af8663725127bd4b

Hash dari panjang nol adalah:
MD5("") = d41d8cd98f00b204e9800998ecf8427e

4. Kesimpulan

Ada beberapa kesimpulan yang didapat melalui pembuatan makalah ini, yaitu:

1. Message Digest 5 merupakan pengembangan lebih lanjut dari Message Digest 4, dimana terjadi penambahan satu ronde
2. Message Digest 5 merupakan sebuah fungsi hash satu arah yang mengubah masukan dengan panjang variabel menjadi keluaran dengan panjang yang tetap, yaitu 128 bit.
3. Keluaran dari Message Digest 5 disebut dengan nilai hash yang berupa 4 buah blok yang masing-masing 32-bit dan akan menjadi 128 bit
4. Message Digest 5 merupakan salah satu solusi untuk menangani persoalan kerahasiaan, autentikasi, keutuhan, dan *non-repudiation* dalam proses mengirim dan menerima pesan atau data.
5. Kecepatan enkripsi pada sistem kriptografi MD5 sangat bergantung kepada spesifikasi komputer yang digunakan
6. Ada 3 buah komponen yang penting dalam proses kriptografi : plainteks, chiperteks, kunci dan algoritma
7. Ada 2 buah proses penting dalam proses kriptografi, yaitu: enkripsi dan deskripsi

REFERENSI

[1] Munir, Rinaldi. (2009). Bahan Kuliah IF2091 Struktur Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Kriptografi.(2009). http://id.wikipedia.org/wiki/Kriptografi Tanggal Akses 8 Desember 2009 Pukul 13.45

[3] Message Digest 5.(2009). http://id.wikipedia.org/wiki/MD5. Tanggal Akses 8 Desember 2009 Pukul 14.05

[4] Hash Function[2009] http://en.wikipedia.org/wiki/Hash_function Tanggal Akses 9 Desember 2009 Pukul 18.00