Transmisi, Vol. 11, No. 1, Juni 2006 : 22 – 27

APLIKASI KRIPTOGRAFI DENGAN ALGORITMA MESSAGE DIGEST 5 (MD5)

Aghus Sofwan, Agung Budi P, Toni Susanto
aghus@elektro.ft.undip.ac.id, agungbp@elektro.ft.undip.ac.id
Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro

Abstrak

Sewaktu kita menerima atau mengirim pesan pada jaringan, terdapat tiga buah persoalan yang sangat penting, kerahasian, autentikasi, keutuhan dan tak berbantahkan (non-repudiation). Message Digest 5 (MD5) adalah salah satu alat untuk 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. Makalah ini mempunyai tujuan untuk merencanakan dan merancang suatu aplikasi untuk menganalisa proses keutuhan atau perubahan pesan dengan menggunakan MD5 dan juga dapat menganalisa hasil keluaran dari MD-5 yang berupa kecepatan dari proses aplikasi yang dibuat.

Kata Kunci : MD5, Kriptografi, hash

1. Latar Belakang

Sewaktu seseorang menerima atau mengirim pesan pada jaringan, terdapat empat buah persoalan yang sangat penting, yaitu kerahasian, autentikasi, keutuhan dan *non repudiation*[1,2,3]. Kerahasian adalah bahwa data kita tidak dapat dibaca oleh orang yang tidak berkepentingan. Autentikasi memberi garansi tentang keaslian data serta dengan siapa kita berhubungan. Keutuhan memberi garansi bahwa data tidak mengalami perubahan sewaktu perjalanan, dengan kata lain data yang dikirim adalah data yang diterima[1,2,3]. Dan *non repudiation* yang berarti si pengirim tidak dapat menyangkal bahwa pesan yang dikirim bukan darinya[2,3].

Salah satu dari bagian kriptografi adalah fungsi hash satu arah. Fungsi hash satu arah adalah dimana kita dengan mudah melakukan enkripsi untuk mendapatkan *cipher*-nya tetapi sangat sulit untuk mendapatkan *plaintext*-nya [1,3,12]. Salah satu fungsi hash yang paling banyak digunakan adalah *Message Digest* 5 (MD-5).

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 variabel 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[10].

Message digest atau intisari pesan harus mempunyai tiga sifat penting, yaitu[3,4]:

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

1.2 Tujuan dan Manfaat

Tujuan yang hendak dicapai adalah merencanakan dan merancang suatu aplikasi untuk menganalisa proses keutuhan atau pun perubahan pesan dengan menggunakan *Message Digest 5* (MD5) dan juga dapat menganalisa hasil keluaran dari MD5 yang berupa kecepatan dari proses aplikasi yang dibuat.

1.3 Batasan Masalah

Untuk tidak memperluas area pembahasan yang terdapat pada proses autentikasi, perlu adanya batasan-batasan untuk menye derhanakan permasalahan, yaitu:

1. Aplikasi hanya memakai MD5, tidak menggunakan enkripsi data.

2. Aplikasi dilakukan pada semua file untuk diubah menjadi ‘intisari pesan’ dengan panjang 128 bit.
3. Contoh Aplikasi hanya simulasi dari fungsi MD5 dalam perubahan pesan

2. Kriptografi dan Fungsi Hash Satu Arah Secara Umum

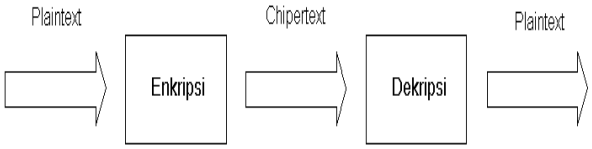
Bagian ini menerangkan tentang kriptografi secara umum yaitu tentang enkripsi dan dekripsi, penggunaan kunci simetrik dan asimetrik, juga tujuan dari kriptografi.

2.1 Prinsip Dasar Kriptografi

Ilmu kriptografi adalah ilmu yang mempelajari tentang penyembunyian huruf atau tulisan sehingga membuat tulisan tersebut tidak dapat dibaca oleh orang yang tidak berkepentingan[2]. Kriptografi sudah dipakai sejak jaman Julius Caesar dimana akan mengirimkan pesan kepada panglimanya tetapi tidak mempercayai kurir pembawa pesan tersebut.

Kriptografi mempunyai 2 (dua) bagian yang penting, yaitu enkripsi dan dekripsi. Enkripsi adalah proses dari penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya. Dekripsi sendiri berarti merubah pesan yang sudah disandikan menjadi pesan aslinya. Pesan asli biasanya disebut *plaintext*, sedangkan pesan yang sudah disandikan disebut *ciphertext*[3].

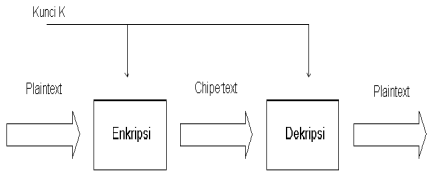
Pada Gambar 2.1 dapat dilihat bahwa masukan berupa *plaintext* akan masuk ke dalam blok enkripsi dan keluarannya akan berupa *ciphertext*, kemudian *ciphertext* akan masuk ke dalam blok dekripsi dan keluarannya akan kembali menjadi *plaintext* semula.



Gambar 2.1 Proses Enkripsi dan Dekripsi

Ada 2 (dua) model algoritma enkripsi yang menggunakan kunci, yaitu kunci simetrik dan kunci asimetrik[2, 3, 12].

Enkripsi kunci simetrik yang biasanya disebut enkripsi konvensional adalah enkripsi yang menggunakan kunci yang sama untuk enkripsi maupun dekripsi, dari Gambar 2.2 terlihat bahwa untuk mengenkripsi maupun mendekripsi pesan hanya menggunakan satu buah kunci (K) saja.



Gambar 2.2 Enkripsi-dekripsi Kunci Simetrik

Penggunaan metode ini membutuhkan persetujuan antara pengirim dan penerima tentang kunci sebelum mereka saling mengirim pesan. Keamanan dari kunci simetrik tergantung pada kerahasiaan kunci, apabila seorang penyusup dapat menemukan kunci maka dengan mudah dapat membaca pesan yang sudah dienkripsi.

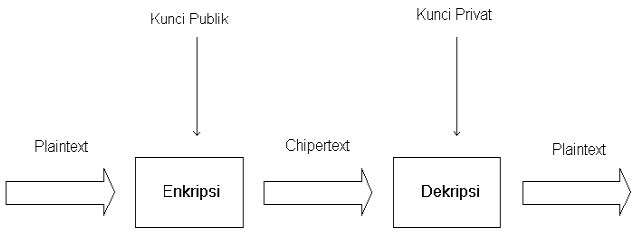
Enkripsi kunci simetrik dapat dibagi kedalam 2 (dua) kelompok yaitu metode *stream cipher* dan metode *block cipher*[3,12].

Enkripsi kunci asimetrik (biasa disebut enkripsi kunci publik) dibuat sedemikian rupa sehingga kunci yang dipakai untuk enkripsi berbeda dengan kunci yang dipakai untuk dekripsi. Enkripsi kunci publik disebut demikian karena kunci untuk enkripsi boleh disebarluaskan kepada umum sedangkan kunci untuk mendekripsi hanya disimpan oleh orang yang bersangkutan[2,3,12]. Enkripsi asimetrik dapat ditulis seperti berikut:

$$E_k(P) = C$$

$$D_k(C) = P$$

Contohnya seperti pada Gambar 2.3 bila seseorang ingin mengirim pesan kepada orang lain maka orang tersebut menggunakan kunci publik orang tersebut untuk mengenkripsi pesan yang kita kirim kepadanya lalu orang tersebut akan mendekripsi pesan tersebut dengan kunci privat miliknya.



Gambar 2.3 Enkripsi Kunci Asimetrik

2.2 Tujuan Dari Kriptografi

Seperti juga perkembangan ilmu kriptografi, tujuan-tujuan dari kriptografi teruslah berkembang. Bila pertama kali dibuat hanya untuk keamanan data saja, tetapi sekarang semakin banyak tujuan-tujuan yang ingin dicapai[12], 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 Prinsip Dasar Fungsi Hash Satu Arah

Fungsi hash satu arah memiliki banyak nama: 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[3].

Fungsi hash satu arah dibuat berdasarkan ide tentang fungsi pemampatan. Fungsi hash adalah sebuah fungsi atau persamaan matematika yang mengambil input dengan panjang variabel (*pre-image*) dan merubahnya menjadi panjang yang tetap (biasanya lebih pendek), keluarannya biasa disebut nilai hash[3,10].

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[3].

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 adalah bila si pengirim (A) akan mengirim pesan kepada temannya (B). Sebelum mengirim, A melakukan hash dari pesannya untuk mendapatkan nilai hash 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 2.4 sehingga secara garis besar, hash dari blok M_i adalah:

$$h_i = f(M_i, h_{i-1})$$

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**[3].

Gambar 2.4 Fungsi Hash Satu Arah

2.4. Sistem Kriptografi MD5

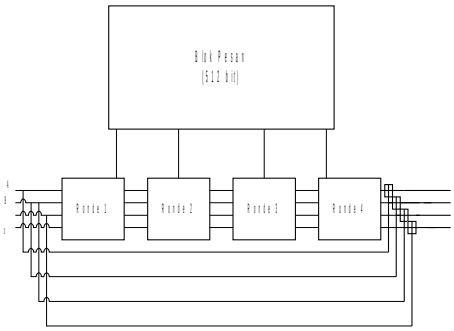
Pada bagian ini dijelaskan mengenai sistem kriptografi MD-5 secara spesifik, yaitu sistem kriptografi algoritma MD5 yang menjelaskan dari awal masukan hingga keluarannya.

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 dan didefinisikan pada RFC 1321[10]. MD-5 merupakan pengembangan dari MD-4 dimana terjadi penambahan satu ronde[1,3,10]. 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[3,10].

Pada Gambar 3.1 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 3.1 simpul utama MD-5

2.4.2 Penjelasan Algoritma MD-5

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[10]. 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.[3,10] . 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[3].**

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 di inisialisasikan 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).

Pada Gambar 3.2 dapat dilihat satu buah operasi dari MD-5 dengan operasi yang dipakai sebagai contoh adalah $FF(a,b,c,d,M_j,s,t_i)$ menunjukan $a = b + ((a + F(b,c,d) + M_j + t_i) <<< s)$

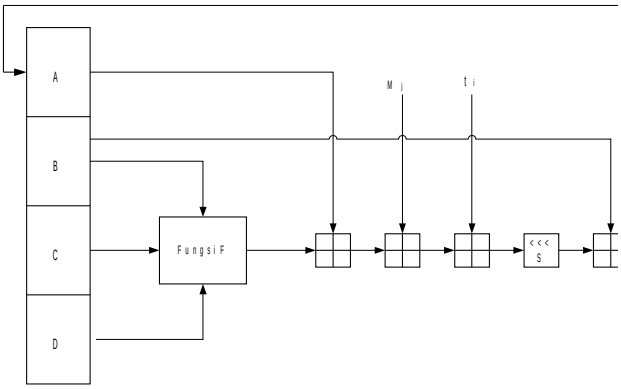
Bila M_j menggambarkan pesan ke-j dari sub blok (dari 0 sampai 15) dan <<<s menggambarkan bit akan digeser ke kiri sebanyak s bit, maka keempat operasi dari masing-masing ronde adalah:

FF(a,b,c,d,M_j,s,t_i) menunjukan a = b + ((a + F(b,c,d) + M_j + t_i) <<< s)

GG(a,b,c,d,M_j,s,t_i) menunjukan a = b + ((a + G(b,c,d) + M_j + t_i) <<< s)

HH(a,b,c,d,M_j,s,t_i) menunjukan a = b + ((a + H(b,c,d) + M_j + t_i) <<< s)

II(a,b,c,d,M_j,s,t_i) menunjukan a = b + ((a + I(b,c,d) + M_j + t_i) <<< s)



Gambar 3.2 Satu buah operasi MD-5

Konstanta t_i 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. Perancangan Aplikasi

- Spesifikasi:

  Aplikasi yang digunakan adalah aplikasi 32-bit yang berjalan pada sistem operasi *Windows* 98 ke atas.

  Aplikasi berbentuk sebuah file *executable* dengan nama **md5_test.exe**

  Aplikasi **md5_string** → mencari nilai hash dari masukan berupa karakter-karakter yang dimasukkan oleh pemakai melalui *keyboard.*

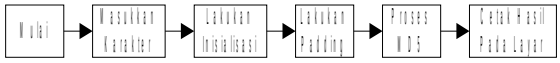
  Aplikasi **md5_file** → mencari nilai hash dari masukan berupa file yang dipilih oleh pemakai.

  Aplikasi ***test suite*** untuk memeriksa bahwa program yang sudah dibuat sudah sesuai RFC 1321.

  Aplikasi ini juga terdapat contoh penggunaan dari MD5.

3.1 Proses MD-5 Dengan Masukan Berupa String

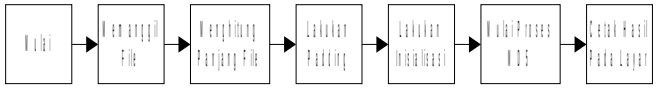
Proses MD5 dengan masukan berupa string adalah proses yang masukan nya berupa karakter-karakter yang dimasukan melalui *keyboard.* Hal ini dapat dilihat pada gambar 3.1



Gambar 3.1 Proses MD5 dengan masukan string

3.2 Proses MD5 Dengan Masukan Berupa File

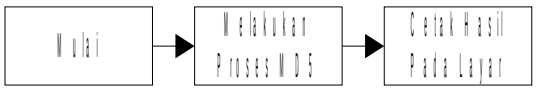
Proses MD5 dengan masukan berupa file adalah proses MD5 yang masukannya memanggil file yang kemudian dihitung berapa panjang bitnya, dalam keadaan ini file diperlakukan sebagai bit memori sehingga masukannya tidak terpengaruh pada ekstensinya. Kemudian dilakukan proses MD5. Hal ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Proses MD5 dengan Masukan File

3.3 Proses MD5 Sebagai Test Suite

Test suite dilakukan untuk mengetahui apakah program yang dibuat ini sudah benar atau tidak. Sebagai perbandingannya digunakan hasil yang sudah dibuat oleh Ron Rivest yang sudah didefinisikan pada RFC 1321. Pada Gambar 3.3 dapat dilihat bahwa masukan dari MD5 sudah ditentukan sehingga hanya membandingkan hasil pada layar dengan yang tercantum pada RFC 1321.



Gambar 3.3 Proses MD5 Sebagai Test Suite

3.4 Contoh Aplikasi

Pada Contoh Aplikasi ini terdapat dua buah aplikasi yang dipakai yaitu Simpan dan Tampil. Simpan akan menyimpan data yaitu berupa Nama, nilai ujian dan nilai hash-nya. Akhir akan melakukan perhitungan MD5 dari data yang disimpan untuk mendapatkan nilai hash-nya yang kemudian membandingkan nilai hash-nya dengan nilai hash dari data semula, apabila nilai hash yang di dapat sama maka data akan ditampilkan. Tetapi bila nilai hash yang di dapat berbeda maka pada form Akhir akan ditampilkan pesan bahwa data terdapat kesalahan atau perubahan

3.5 Pengukuran Kecepatan Aplikasi

Pengukuran kecepatan aplikasi merupakan sebuah analisa yang akan dipakai untuk mengukur tingkat kecepatan dari proses mencari nilai hash dari file dengan menggunakan aplikasi MD5

Adapun rumus yang dipakai dalam aplikasi untuk menghitung kecepatan mencari nilai hash tersebut adalah:

$$\text{Kecepatan} = \frac{\text{Besar Ukuran File}}{\text{Lama waktu proses}}$$

Satuan dari kecepatan enkripsi ini adalah Mbytes/detik

Dalam analisa kecepatan ini, akan dilakukan sebanyak 5 (lima) kali pengambilan waktu terbaik yang diperlukan untuk enkripsi untuk setiap filenya kemudian dicari waktu rata-ratanya.
besar file dan tabel perbandingan kecepatan maksimum dengan kecepatan rerata terhadap besar file

4. Analisis Kecepatan MD5

Analisis kecepatan disini adalah analisis tentang kecepatan aplikasi dalam mengenkrip file untuk mencari nilai hash. Analisis dilakukan untuk mencari kecepatan aplikasi dengan masukan file yang mempunyai besar berbeda-beda

Pengujian dilakukan dengan cara mengenkrip file sebanyak 31 (tigapuluh satu) buah file dengan besar file yang berbeda-beda. Setiap file dilakukan pengambilan waktu eksekusi sebanyak 5 kali kemudian mencari waktu reratanya.

4.1 Hasil Pengujian

Hasil pengujian digambarkan dengan tabel hasil pengujian, yang kemudian dijabarkan dengan grafik hasil uji coba terhadap file yaitu Grafik kecepatan aplikasi terhadap besar file, Grafik rerata waktu eksekusi terhadap besar file dan Tabel perbandingan kecepatan maksimum dengan kecepatan rerata terhadap besar file.

5 Kesimpulan

Beberapa kesimpulan yang dapat diperoleh dari makalah ini adalah:

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. Rerata kecepatan dari program aplikasi MD-5 adalah 7,1633 Mbytes/detik
3. Aplikasi yang dibuat hanya efektif digunakan untuk ukuran file kurang dari 40 Mbytes.
4. Sumber daya komputer berpengaruh terhadap kecepatan enkripsi.

6. Saran

Adapun saran-saran dari penulis tentang aplikasi ini adalah:

1. Aplikasi MD5 dapat digabungkan dengan algoritma kriptografi lainnya sehingga didapat aplikasi kriptografi yang lebih handal.
2. Untuk pengembangan selanjutnya, contoh aplikasi dapat mempergunakan 2 (dua) komputer atau lebih.

7. DAFTAR PUSTAKA

1. A.Menezes, P. Van Oorschot, S. Vanstone. Handbook of applied Cryptography. CRC Press 1996.
2. Pfleeger, Charles P, Security in Computing Second Edition. Prentice-Hall International, Inc, New Jersey, 1997
3. B. Schneier, Applied Cryptography, John Wiley & Sons, New York, 1994
4. Tanenbaum, Andrew S, Jaringan Komputer Edisi Indonesia Dari Computer Network Edisi III, Prenhallindo, Jakarta, 1997
5. ---, http:// www.counterpane.com/
6. ---, http:// www.cryptography.com.
7. ---, http:// www.cryptography.org
8. ---,http://www.eskino.com/~weidai /benchrarks.html.
9. ---, http:// www.cwi.nl/~kik/persb-UK.html.
10. ---, http://www.faqs.org/ftp/rfc/ rfc1321.txt
11. ---, http://www.spitzner.net/pubs.html
12. ---, http://www.secure-hash-algorithm-md5-sha-1.co.uk/index.htm