**BAB I**

**PENDAHULUAN**

1. Latar Belakang

Perkembangan teknologi informasi dewasa ini telah berpengaruh pada hampir semua aspek kehidupan manusia. Saat ini internet seolah-olah menghiraukan batas-batas geografis suatu negara, bukan hal yang sulit bagi seseorang untuk berkomunikasi jarak jauh, mengirimkan data, mencari informasi dan sebagainya. Semua hal tersebut dapat dilakukan dengan internet secara cepat, efisien dan relatif murah. Namun di sisi lain, ternyata internet merupakan jalur yang tidak terlalu aman karena merupakan media komunikasi umum yang dapat digunakan secara bebas oleh siapapun sehingga sangat rawan penyadapan informasi oleh pihak-pihak yang tidak absah. Oleh karena itu masalah keamanan, kenyamanan dan otorisasi dalam berinternet menjadi masalah krusial yang sampai hari ini masih menjadi perbincangan hangat. Banyak pertanyaan yang muncul, Apakah transaksi via internet bisa terjamin keamanannya? Apakah data yang dikirimkan via internet sampai pada tujuan yang tepat tanpa diketahui pihak yang tidak diinginkan? Bagaimana mencegah penyadapan informasi? Bagaimana mengantisipasi seseorang mencuri data?

Banyak metode yang dapat dilakukan untuk menjawab pertanyaan-pertanyaan diatas. Salah satunya dengan metode penyandian atau yang lebih dikenal dengan metode kriptografi. Banyak sekali metode penyandian atau metode kriptografi yang dikembangkan oleh pakar-pakar kriptografi hingga saat ini. Hal ini dilakukan dikarenakan penyadap dan pencuri informasi atau yang lebih dikenal dengan sebutan cracker semakin handal dalam mempenetrasi suatu sistem untuk menggali berbagai macam informasi. Oleh karena itu dalam rangka melawan tindakan keji tersebut, pakar-pakar kriptografi terus mengembangkan metode ini secara berkesinambungan. Banyak sekali metode penyandian yang telah diciptakan oleh pakar-pakar kriptografi dunia, sebut saja Algoritma DES, Algoritma 3DES, Algoritma IDEA, Algoritma Blowfish, Algotitma RSA, Algoritma MD4, Algotitma MD5, Algoritma SHA-1, Algoritma McEliecce dan sebagainya. Algoritma-algoritma diatas telah diuji kemampuannya oleh pakar-pakar kriptografi, namun tidak semua metode kriptografi diatas bertahan dari serangan para penyadap informasi atau dalam istilah kriptografi sering disebut dengan cryptanalist. Sebut saja algoritma DES, sempat bertahan cukup lama yakni selama 20 tahun akhirnya harus rela di-crack hanya dalam tempo 3,5jam dengan biaya 1 juta US Dollar pada tahun 1993. Namun para pakar kriptografi tidak berdiam diri, karena setelah kejadian itu para pakar kriptograhi mengembangkan varian baru dari algoritma DES seperti Algoritma 3DES, Algoritma RDES dan sebagainya.

Makalah ini akan membahas salah satu diantara banyak kriptografi handal yang hingga saat ini masih cukup kuat membendung serangan-serangan dari para cryptanalist, yaitu metode kriptografi dengan algoritma Message Digest 5 (MD5). Selain karena metode kriptografi ini belum mampu di patahkan oleh para cryptanalist, makalah ini membahas kriptografi MD5 karena algoritma ini merupakan varian baru setelah algoritma MD2 dan algoritma MD4 sehingga dirasa cukup up to date untuk dibahas saat ini.

1. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah di uraikan, dapat dirumuskan masalah-

masalah yang akan dibahas pada penulisan kali ini. Masalah yang dimaksud adalah

sebagai berikut :

1. Bagaimana sejarah kriptografi.
2. Apakah tujuan kriptografi.
3. Apakah pengertian fungsi hash.
4. Bagaimana sejarah MD5.
5. Bagaimana cara kerja MD5.
6. Bagaimana Aplikasi Algoritma MD5 Untuk Integritas Data.
7. Tujuan Pembuatan Makalah

Pembuatan makalah bermaksud untuk menganalisa proses penyandian pesan dan proses membuka kembali penyandian dalam kriptografi, khususnya kriptografi algoritma MD5. Diharapkan dengan adanya makalah ini para pengguna jalur internet lebih ekstra hati-hati dalam penggunaan jalur publik ini. Untuk metode pengamanan dapat menggunakan metode kriptografi algoritma MD5 yang dibahas secara detail pada makalah ini.

**BAB II**

**LANDASAN TEORI**

1. Sejarah Kriptografi

Dalam The CodeBreaker yang ditulis oleh Kahn, terlihat bahwa kriptografi mempunyai sejarah yang panjang. Kriptografi sudah digunakan oleh bangsa Mesir Kuno sekitar 4000 tahun sampai abad 20 dimana kriptografi berperanan penting di dalam perang dunia pertama dan kedua. Juga diungkap bahwa latar belakang sejarah yang panjang itu sangat menentukan perkembangan ilmu kriptografi itu sendiri baik dari segi teoretik maupun aplikasinya.

Awalnya kriptografi sangat dominan digunakan dalam bidang-bidang yang berhubungan dengan militer, layanan diplomatik, dan pemerintahan secara umum. Dalam hal ini kriptografi digunakan sebagai suatu alat untuk melindungi strategi dan rahasia negara. Perkembangan sistem komunikasi dan komputer pada tahun 1960 an membawa kriptografi memasuki sektor swasta sebagai alat untuk melindungi informasi dalam bentuk dijitel dan untuk memberikan layanan keamanan.

Hasil kerja Feistel di IBM pada awal tahun 1970 an dan puncaknya ada tahun 1977, DES (Data Ecryption Standart) merupakan karya kriptografi yang paling terkenal di dalam sejarah. Karya ini menjadi alat keamanan komersial elektronik di banyak institusi keuangan di seluruh dunia hingga pertengahan tahun 1990-an. DES secara definitif terbukti tak-aman sejak Juli 1998. Walaupun demikian DES telah melandasi prinsip-prinsip sandi simetrik modern yang dewasa ini muncul produk-produk penggantinya seperti: AES (Advanced Ecryption Standart), Blowfish, 3DES, RC5, dan lain sebagainya.

Yang cukup signifikan selanjutnya adalah pada tahun 1976 ketika Diffie dan Hellman mempublikasikan suatu artikel dengan judul New Directions in Cryptography. Artikel ini memperkenalkan konsep revolusioner tentang kriptograpy kunci-publik (public-key cryptography) dan juga memberikan suatu metode baru untuk perubahan kunci dimana keamanan didasarkan pada pemecahan problem logaritme diskret. Walaupun penulis pada saat itu mengungkapkan hanya segi teoretiknya tanpa bentuk praktisnya, namun karya ini telah memberikan cakrawala baru bagi para ilmuwan kriptografi. Ini terbukti pada tahun 1978, Rivest, Shamir, dan Adleman menemukan bentuk praktis yang pertama untuk skema enkripsi dan penandaan kunci publik yang sekarang dikenal dengan skema RSA. Skema ini didasarkan pada problem matematika yang sulit lainnya, yaitu pemecahan masalah faktorisasi intejer besar. Bentuk praktis skema kunci-publik lainya ditemukan oleh ElGamal pada tahun 1985. Sebagaimana karya Di¢ e dan Hellman, skema ini juga didasarkan pada pemecahan problem logaritme diskret.

Sumbangan yang paling signifikan yang diberikan olen kriptografi kunci-publik adalah penandaan dijitel (digital signature). Pada tahun1991 standar internasional pertama untuk penandaan dijitel diadopsi dari ISO/IEC 9796. Standar internasional penandaan dijitel ini didasarkan ada skema kunci-publik RSA. Pada tahun 1994 pemerintah Amerika Serikat mengadopsi standar penandaan dijitel yang mekanismenya didasarkan pada skema kunci-publik ElGamal.

1. Tujuan Kriptografi

Secara umum tujuan ilmu kriptografi diciptakan adalah untuk merahasiakan suatu pesan agar tidak bisa dibaca oleh pihak yang tidak absah. berpedoman dari tujuan umum tersebut, terdapat empat tujuan mendasar ilmu kriptografi digunakan dalam bidang keamanan informasi yaitu :

1. Kerahasiaan

adalah layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

1. Integritas data

adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

1. Autentikasi

adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

1. Non-repudiasi

adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

**BAB III**

**PEMBAHASAN**

1. Pengertian Fungsi Hash

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:

* 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).
* 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.

1. Sejarah MD5

MD5 adalah salah satu dari serangkaian algortima(merupakan salah satu fungsi Hash) message digest yang didesain oleh Profesor Ronald Rivest dari MIT (Rivest, 1994). Saat kerja analitik menunjukkan bahwa pendahulu MD5, yaitu MD4 mulai tidak aman, MD5 kemudian didesain pada tahun 1991 sebagai pengganti dari MD4 (kelemahan MD4 ditemukan oleh Hans Dobbertin).

Pada tahun 1993, den Boer dan Bosselaers memberikan awal, bahkan terbatas, hasil dari penemuan pseudo-collision dari fungsi kompresi MD5. Dua vektor inisialisasi berbeda I dan J dengan beda 4-bit diantara keduanya.

MD5compress(I,X) = MD5compress(J,X)

Pada tahun 1996 Dobbertin mengumumkan sebuah kerusakan pada fungsi kompresi MD5. Dikarenakan hal ini bukanlah serangan terhadap fungsi hash MD5 sepenuhnya, hal ini menyebabkan para pengguna kriptografi menganjurkan pengganti seperti WHIRLPOOL, SHA-1 atau RIPEMD-160.

Ukuran dari hash — 128-bit — cukup kecil untuk terjadinya serangan brute force.. MD5CRK adalah proyek distribusi mulai Maret 2004 dengan tujuan untuk menunjukkan kelemahan dari MD5 dengan menemukan kerusakan kompresi menggunakan brute force attack. Bagaimanapun juga, MD5CRK berhenti pada tanggal 17 Agustus 2004, saat [[kerusakan hash]] pada MD5 diumumkan oleh Xiaoyun Wang, Dengguo Feng, Xuejia Lai dan Hongbo Yu. Serangan analitik mereka dikabarkan hanya memerlukan satu jam dengan menggunakan IBM P690 cluster.

Pada tanggal 1 Maret 2005, Arjen Lenstra, Xiaoyun Wang, and Benne de Weger mendemontrasikan kunstruksi dari dua buah sertifikat X.509 dengan public key yang berbeda dan hash MD5 yang sama, hasil dari demontrasi menunjukkan adanya kerusakan. Konstruksi tersebut melibatkan private key untuk kedua public key tersebut. Dan beberapa hari setelahnya, Vlastimil Klima menjabarkan dan mengembangkan algortima, mampu membuat kerusakan Md5 dalam beberapa jam dengan menggunakan sebuah komputer notebook. Hal ini menyebabkan MD5 tidak bebas dari kerusakan.

Dikarenakan MD5 hanya menggunakan satu langkah pada data, jika dua buah awalan dengan hash yang sama dapat dibangun, sebuah akhiran yang umum dapat ditambahkan pada keduanya untuk membuat kerusakan lebih masuk akal. Dan dikarenakan teknik penemuan kerusakan mengijinkan pendahuluan kondisi hash menjadi arbitari tertentu, sebuah kerusakan dapat ditemukan dengan awalan apapun. Proses tersebut memerlukan pembangkitan dua buah file perusak sebagai file template, dengan menggunakan blok 128-byte dari tatanan data pada 64-byte batasan, file-file tersebut dapat mengubah dengan bebas dengan menggunakan algoritma penemuan kerusakan.

Saat ini dapat diketahui, dengan beberapa jam kerja, bagaimana proses pembangkitan kerusakan MD5. Yaitu dengan membangkitkan dua byte string dengan hash yang sama. Dikarenakan terdapat bilangan yang terbatas pada keluaran MD5 (2128), tetapi terdapat bilangan yang tak terbatas sebagai masukannya, hal ini harus dipahami sebelum kerusakan dapat ditimbulkan, tapi hal ini telah diyakini benar bahwa menemukannya adalah hal yang sulit.

Sebagai hasilnya bahwa hash MD5 dari informasi tertentu tidak dapat lagi mengenalinya secara berbeda. Jika ditunjukkan informasi dari sebuah public key, hash MD5 tidak mengenalinya secara berbeda jika terdapat public key selanjutnya yang mempunyai hash MD5 yang sama.

Bagaimanapun juga, penyerangan tersebut memerlukan kemampuan untuk memilih kedua pesan kerusakan. Kedua pesan tersebut tidak dengan mudah untuk memberikan serangan preimage, menemukan pesan dengan hash MD5 yang sudah ditentukan, ataupun serangan preimage kedua, menemukan pesan dengan hash MD5 yang sama sebagai pesan yang diinginkan.

Hash MD5 lama, yang dibuat sebelum serangan-serangan tersebut diungkap, masih dinilai aman untuk saat ini. Khususnya pada digital signature lama masih dianggap layak pakai. Seorang user boleh saja tidak ingin membangkitkan atau mempercayai signature baru menggunakan MD5 jika masih ada kemungkinan kecil pada teks (kerusakan dilakukan dengan melibatkan pelompatan beberapa bit pada bagian 128-byte pada masukan hash) akan memberikan perubahan yang berarti. Penjaminan ini berdasar pada posisi saat ini dari kriptoanalisis. Situasi bisa saja berubah secara tiba-tiba, tetapi menemukan kerusakan dengan beberapa data yang belum-ada adalah permasalahan yang lebih susah lagi, dan akan selalu butuh waktu untuk terjadinya sebuah transisi.

1. Cara Kerja MD5

Algoritma MD5 adalah algoritma yang menggunakan fungsi hash satu arah yang diciptakan oleh Ron Rivest. Algoritma merupakan pengembangan dari algoritma-algoritma sebelumnya yaitu algoritma MD2 dan algoritma MD4 karena kedua algoritma ini berhasil diserang para *cryptanalist*.

Cara kerja kriptografi algoritma MD5 adalah menerima input berupa pesan dengan ukuran sembarang dan menghasilkan message diggest yang memiliki panjang 128 bit. Berikut ilustrasi gambar dari pembuatan message diggest pada kriptografi algoritma MD5 :



Gambar 3.1 Pembuatan message digest dengan algoritma MD5

Menilik dari gambar diatas, secara garis besar pembuatan message digest ditempuh melalui empat langkah, yaitu :

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 448 modulo 512. Ini berarti setelah menambahkan bit-bit pengganjal, kini panjang pesan adalah 64 bit kurang dari kelipatan 512. Hal yang perlu diingat adalah angka 512 muncul karena algoritma MD5 memproses pesan dalam blok-blok yang berukuran 512.

Apabila terdapat pesan dengan panjang 448 bit, maka pesan tersebut akan tetap ditambahkan dengan bit-bit pengganjal. Pesan akan ditambahkan dengan 512 bit menjadi 96 bit. Jadi panjang bit-bit pengganjal adalah antara 1 sampai 512. Lalu satu hal lagi yang perlu diperhatikan adalah bahwasanya bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

1. Penambahan nilai panjang pesan semula

kemudian proses berikutnya adalah pesan ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Apabila panjang pesan lebih besar dari 264 maka yang diambil adalah panjangnya dalam modulo 264. dengan kata lain, jika pada awalnya panjang pesan sama dengan K bit, maka 64 bit yang ditambahkan menyatakan K modulo 264. sehingga setelah proses kedua ini selesai dilakukan maka panjang pesan sekarang adalah 512 bit.

1. Inisialisasi penyangga MD

Pada algoritma MD5 dibutuhkan empat buah penyangga atau buffer, secara berurut keempat nama penyangga diberi nama A, B, C dan D. Masing-masing penyangga memiliki panjang 32 bit. Sehingga panjang total :



Keempat penyangga ini menampung hasil antara dan hasil akhir. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi Hexadesimal) sebagai berikut :

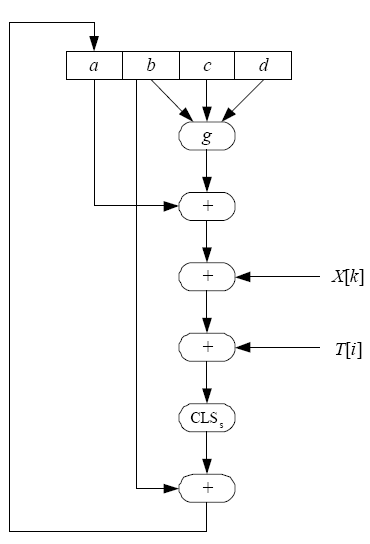


1. Pengolahan pesan dalam blok berukuran 512 bit

Proses berikutnya adalah pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y0 sampai YL-1). Setelah itu setiap blok 512 bit diproses bersama dengan penyangga MD yang menghasilkan keluaran 128 bit, dan ini disebut HMD5. Berikut ini gambaran dari proses HMD5 :

Gambar 3.2 Pengolahan blok 512 bit (Proses HMD5)

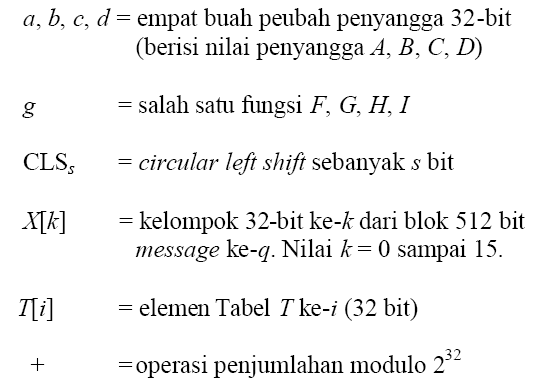
Dari gambar diatas dapat kita lihat bahwa proses HMD5 terdiri dari 4 buah putaran, dan masing-masing putaran melakukan opersi dasar MD5 sebanyak 16 kali. Dimana disetiap operasi dasar memakai sebuah elemen T. Sehingga setiap putaran memakai 16 elemen tabel T.

Pada gambar 3.2, Yq menyatakan blok 512 bit ke-q dari pesan yang telah ditambahkan dengan bit-bit pengganjal pada proses pertama dan tambahan 64 bit nilai panjang pesan semula pada proses kedua. MDq adalah nilai *message digest* 128 bit dari proses HMD5 ke-q. Pada awal proses , MDq berisi nilai inisialisasi penyangga MD. Kemudian fungsi fF, fG, fH, dan fI pada gambar, masing-masing berisi 16 kali operasi dasar terhadap input, setiap operasi dasar menggunakn elemen tabel T. Berikut ini ilustrasi gambar operasi dasar MD5 :

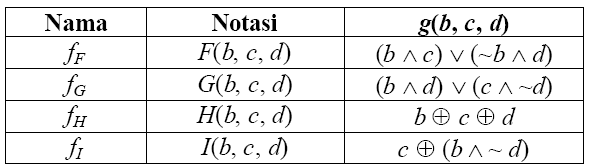
Gambar 3.3 Operasi dasar MD5

Operasi dasar MD5 yang diperlihatkan gambar diatas dapat dituliskan dengan persamaan berikut ini :

https://lh3.googleusercontent.com/yc1r6MUqLhvl51nGPUE-Rv5PUhO24HhD5F0jrYruf-7VTGyBPSu-ym-Qy8ogAS3LXakZf-AMHZ_rPlVJVf-lYQ4YnwGb6xeMpHmfJi_grcO8yi_Hknjyse1ArY7CFa4EDHaRF5Pxoeeup1mplg

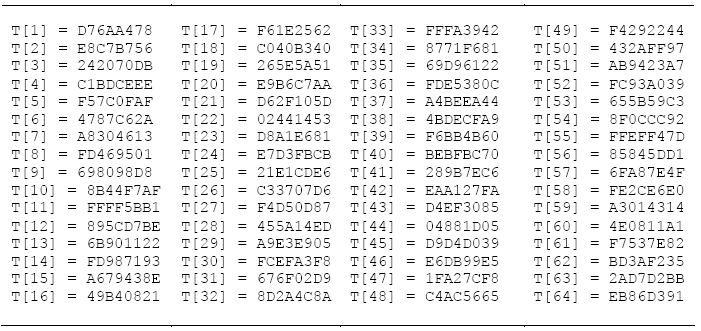
Dimana :

Fungsi fF, fG, fH, dan fI adalah fungsi untuk memanipulasi masukan a, b, c, dan d dengan ukuran 32-bit. Masing-masing fungsi dapat dilihat pada Tabel 1 dibawah ini :

Tabel 1 Fungsi-fungsi dasar MD5

NB: secara berurut peartor logika AND, OR, NOT dan XOR dilambangkan dengan ∧, ∨, ~, ⊕

Kemudian nilai T[i] dapat dilihat pada tabel dibawah ini. Tabel ini disusun oleh fungsi 232 x abs(sin(i)), i dalam radian.

Tabel 2. Nilai T[i]

Sebagaimana telah dijelaskan sebelumnya bahwa fungsi fF, fG, fH, dan fI melakukan 16 kali operasi dasar. Misalkan notasi berikut ini :

https://lh4.googleusercontent.com/KNYqQeuE7ee-uetp4ZzmTtRqVgftNTzZy211ck5GzAYW_1_4tMhQMI9wG3KR-igME2CAhzRDgnGkJACrAtM_w67L7xP8X40BzEdxfqcaGi4BiXtjXuQet5-ACWk8j80Q-7_FiJwRQmN_K8bmMg

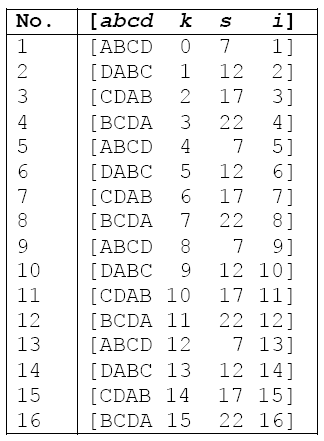
Meyatakan operasi

https://lh3.googleusercontent.com/3iFGid5GP_vunvEXdR1lGEOdFFKzIBDOUBO9X9Oi11mPpYIsFjjCwRqpfpBEpgLfYa-UIw6DqEe7YNzP7zQX1DER8SqeY6-Bk0yyuSynQ9_JmvaVvcDMFD01tvB1KfbefsmB4GqhcHsZdGHi8A

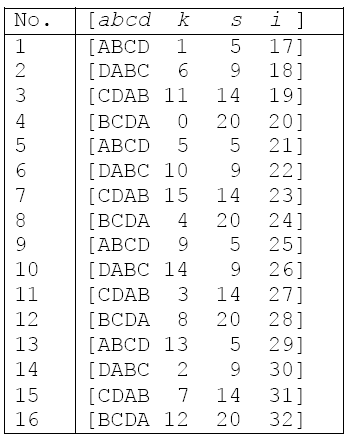
untuk operasi diatas, <<<s melambangkan opersi *circular left shift* 32 bit, maka operasi dasar pada masing-masing putaran dapat ditabulasikan sebagai berikut :

* putaran 1 : 16 kali operasi dasar dengan g(b, c, d) – F(b, c, d), dapat dilihat pada tabel berikut ini :

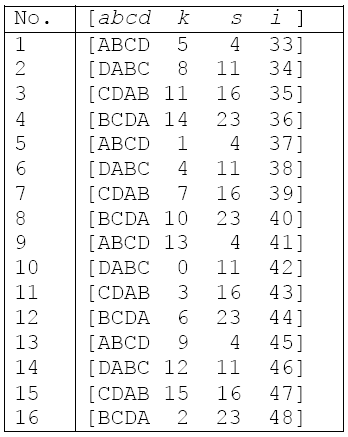
Tabel 3 Rincian operasi pada fungsi F(b, c, d)



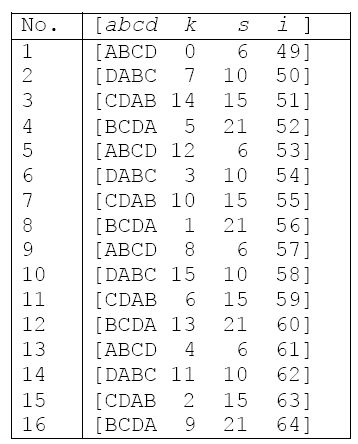
* putaran 2 : 16 kali operasi dasar dengan g(b, c, d) – G(b, c, d), dapat dilihat pada tabel berikut ini :

Tabel 4 Rincian operasi pada fungsi G(b, c, d)

* putaran 3 : 16 kali operasi dasar dengan g(b, c, d) – H(b, c, d), dapat dilihat pada tabel berikut ini

Tabel 5 Rincian operasi pada fungsi H(b, c, d)

* putaran 4 : 16 kali operasi dasar dengan g(b, c, d) – I(b, c, d), dapat dilihat pada tabel berikut ini :

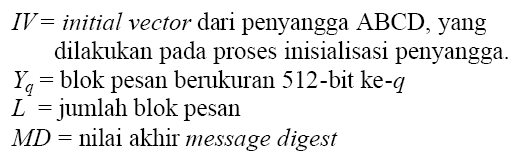
Tabel 6 Rincian operasi pada fungsi I(b, c, d)

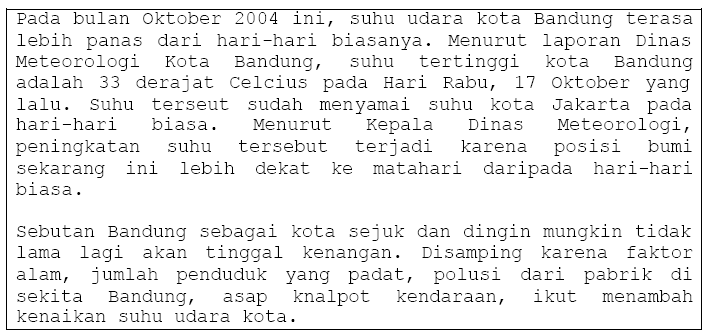
Setelah putaran keempat, a, b, c dan d di tambahkan ke A, B, C dan D yang selanjutnya algoritma akan memproses untuk blok data berikutnya (Yq+1). Output akhir dari algoritma MD5 adalah hasil penyambungan bit-bit di A, B, C dan D.

https://lh4.googleusercontent.com/TP3lYN8Ih19TPMiWo1LpuM_Y9NLRvsotnRzJKeryDgTUdhSQtGt6ZiKhK5KGCU5iaI3DkkRbsCpuOD3gUBtEiRHfqGajQctj8pySqPmmzdt6JZLF70Z_6_EvNeNnNfwwF9TcnsUPAUaDIN2fwQDari uraian diatas, secara umum fungsi hash MD5 dapat ditulis dalam persamaan matematis sebagai berikut :

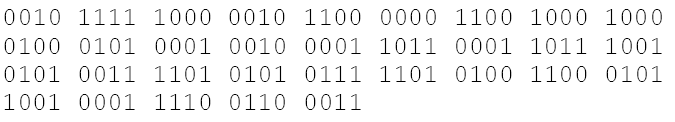
https://lh6.googleusercontent.com/fE3htJqzjmPU1ReBuwuV1-YmDqd378MA37gabrJWfSvtpbRsEDsdnjtF4q_LHGJxMQWUXfvrTVh-OyZD5-PSQHit2etBAhr1Ka9jyq0K7y0PY7I2xjErwA4bk91IwstIuf1LsKbpeL344VTvGg

https://lh6.googleusercontent.com/K7bVR59pKvc-YIIUU0oKrxmw0aF6KLlJiRVhDGuP_PA3NFYYgDo7EwZGj0gSEYhzuWx1m0-tRPOyubtNmuta7_1opt7KAqsrloHYBcWIdOdjBYUCg2ZwlaF_YkTz5mrmMCG4iBM5yYwsXJo3HQ

Dimana

Agar lebih mudah dipahami, berikut ini contoh penerapan kriptografi MD5 pada suatu pesan yang ingin dirahasiakan. Misalkan terdapat sebuah arsip dengan nama bandung.txt sebagai berikut :

Message digest yang dihasilkan dari arsip diatas adalah untuk 128 bit :

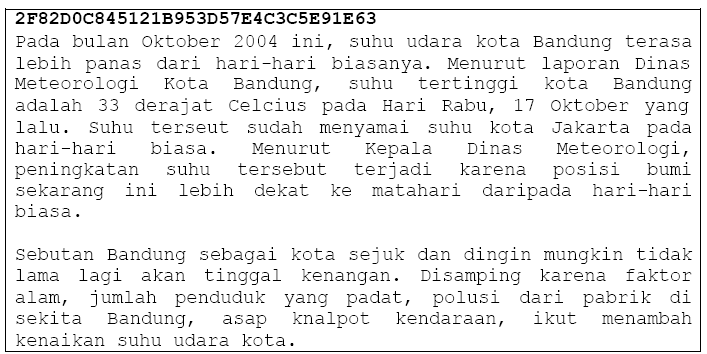


Lalu untuk notasi hexadesimal :

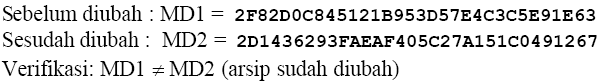
https://lh6.googleusercontent.com/3fEIyvxnVCkaeGo5n1x7sWPueH3dJkoI6ZvcBwu6TE1dRB2Wjdy-ow_WSVIwOUSQIycmeHlYf5TiwlAn37CDHP4n2Lumw9ufwNL-QUXPRDAEFQEksRFHkEBPBHqtbxanXVzeRyS_4r6LbGdSuA

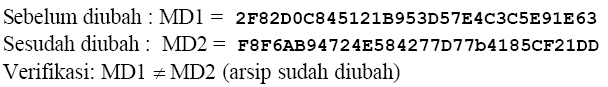
1. Aplikasi Algoritma MD5 Untuk Integritas Data

Aplikasi algoritma dapat digunakan untuk menjaga keintegritasan data. Dengan aplikasi fungsi hash yang menjadi asas algoritma MD5, perubahan kecil pada data sekalipun dapat terdeteksi. Langkah yang harus ditempuh adalah bangkitkan *message digest* dari isi arsip menggunakan algoritma MD5. kemudian gabung *message digest* ke dalam arsip. Verifikasi isi arsip dapat dilakukan secara berkala dengan membandingkan MD isi arsip sekarang dan MD dari arsip asli (MD yang telah disimpan sebelumnya). Jika hasilnya berbeda maka telah terjadi perubahan pada arsip.

Aplikasi ini didasarkan pada kenyataan bahwa perubahan 1 bit pesan akan mengubah, secara rata-rata, setengah dari bit-bit *message digest*. Dengan kata lain, algoritma MD5 dengan fungsi hashnya sangat peka terhadap perubahan sekecil apapun pada data masukan. Sebagai contoh perhatikan arsip dibawah ini :

Untuk kasus pertama, apabila 33 derajat celcius diganti dengan 32, maka MD dari isi arsip adalah (tidak termasuk baris MD) :



Untuk kasus kedua, apabila ditambahkan spasi diantara ”33” dan ”derajat”, maka MD dari isi arsip adalah (tidak termasuk baris MD) :

Dari kedua kasus diatas sangat terlihat bahwa algoritma MD5 dengan fungsi hashnya sangatlah peka dengan perubahan walau kecil sekalipun.

**BAB IV**

**PENUTUP**

1. Kesimpulan

Dari pembahasan pada bab-bab sebelumnya maka penulis dapat menarik kesimpulan sebagai berikut:

1. Algoritma *Message Digest 5* (MD5) adalah algoritma yang dikembangkan setelah algoritma MD2 dan MD4.
2. Algoritma *Message Digest 5* (MD5) adalah sebuah fungsi hash satu arah yang mengubah masukan dengan panjang variabel menjadi keluaran dengan panjang tetap yaitu 128 bit.
3. Algoritma MD5 dengan fungsi hashnya sangat peka terhadap perubahan pesan, maka algoritma MD5 cocok untuk aplikasi yang menjaga integritas suatu data.