BAB II

LANDASAN TEORI

2.1 KEAMANAN KOMPUTER

Kata keamanan dalam dunia komputer adalah hal yang sangat diidam-idamkan oleh para pemakai komputer. Hal ini dikarenakan oleh mudahnya sebuah komputer menjadi target sasaran orang-orang yang tidak bertanggung jawab untuk diubah isinya. Apalagi sekarang dengan makin maraknya internet, maka orang yang tidak bertanggung jawab bisa merusak komputer atau server lain dari jarak jauh.

Berbagai macam cara dilakukan untuk mengamankan data-data yang sangat bernilai, seperti memasang *software* anti virus, firewall, melakukan enkripsi data dan lain-lain.

2.1.1 Validasi

Salah satu cara untuk mengamankan komputer atau data adalah dengan melakukan validasi. Validasi berarti melakukan pemeriksaan terhadap suatu hal untuk ditentukan apakah hal tersebut benar atau salah sesuai dengan standar atau batasan yang ditentukan. Standar atau batasan yang dimaksud disini adalah suatu standar yang ditentukan sendiri oleh perancang sistem. Semakin ketat standar yang dipakai, maka program atau data akan semakin aman.

Contoh penggunaan validasi adalah pada program komputer yang berfungsi untuk memasukkan data ke database. Dalam program tersebut diasumsikan batasan data yang boleh dimasukan adalah dengan tipe angka, maka program atau sistem tentu akan menolak memasukan data dengan tipe karakter.

2.2 MD5

2.2.1 Message Digest

Dalam situs *ietf.org rfc rfc1321.txt* dinyatakan bahwa *message digest* adalah sebuah tanda tangan digital yang padat untuk suatu data atau dokumen yang memiliki data biner. Suatu algoritma tanda tangan digital yang baik, tidak akan menghasilkan tanda tangan digital yang sama untuk *input* yang berbeda. Tetapi untuk memenuhi teori tersebut, akan memerlukan suatu tanda tangan digital yang panjangnya sama dengan panjang data *input*.

Algoritma *message digest* memiliki banyak persamaan dengan teknik yang digunakan pada enkripsi, namun dengan hasil yang berbeda. Enkripsi mengubah isi dokumen menjadi kode-kode yang tidak dimengerti manusia yang tidak berhak mengetahuinya, dan bisa diubah kembali ke bentuk aslinya dengan memakai kunci dekripsi. Sedangkan *message digest* menghasilkan tanda tangan digital, yang merupakan hasil perhitungan dari data *string* yang diinputkan, tetapi tanda tangan digital tersebut tidak bisa diubah kembali menjadi *string input*.

2.2.2 Sekilas MD5

MD5 adalah algoritma *message digest* 128 bit yang dibuat oleh Professor Ronald L. Rivest dari Massachusetts Institute of Technology (MIT) dan dipublikasikan pada bulan April 1992. Professor Ronald Rivest menyatakan bahwa algoritma MD5 akan menghasilkan tanda tangan digital 128 bit dari suatu input, tidak peduli berapapun panjangnya.

Secara sederhana bisa dinyatakan algoritma MD5 melakukan "kompresi" terhadap suatu input, baik panjang maupun pendek, yang hasilnya adalah tanda tangan digital sepanjang 32 (tiga puluh dua) karakter [RIV91].

MD5 merupakan bantahan atas teori yang menyatakan, untuk menghasilkan tanda tangan digital yang baik maka panjang tanda tangan digital harus sama dengan panjang masukannya. Berikut ini adalah contoh tanda tangan digital dengan menggunakan algoritma MD5.

```
1. md5 ("B") = 0947f85161b05919d96940f3de14852e
2. md5 ("b") = 92eb5ffee6ae2fec3ad71c777531578f
3. md5 ("a") = 0cc175b9c0f1b6a831c399e269772661
4. md5 ("a.") = 9fbcccf456ef61f9ea007c417297911d
5. md5 ("a ") = 99020cb24bd13238d907c65cc2b57c03
6. md5 ("a  ") = d4ac0334c4130de05b4a37a87590ccc4
7. md5 ("a,") = 3ded2184a3e467984dba5788f82cc430
```

Contoh pertama menunjukkan hasil output karakter "B". Contoh kedua adalah output karakter "b". Ternyata dari hasil perbandingan terlihat bahwa walaupun terlihat hampir sama, tetapi jenisnya berbeda maka fungsi MD5 akan mengeluarkan hasil yang tidak identik. Lima contoh terakhir menunjukkan bahwa walaupun huruf yang diinputkan sama, tetapi penambahan karakter atau spasi sebanyak satu atau dua spasi serta perubahan apapun terhadap input akan memberikan output berbeda.

Dari contoh diatas dapat disimpulkan bahwa algoritma MD5 selalu menghasilkan tanda tangan digital sepanjang 32 karakter, tanpa tergantung panjang input. Selain itu hasil output tidak akan sama untuk input yang berbeda [LIO03].

Algoritma MD5 sendiri sudah dimasukkan sebagai fungsi bawaan dari beberapa bahasa pemrograman seperti C, Java, Perl, PHP. Cara pemanggilan dan menampilkan MD5 pada PHP adalah:

```
<?
        md5 (" <string> ");
?>
```

2.2.3 Alasan Pemilihan MD5

Algoritma MD5 digunakan dalam penelitian tugas akhir ini dikarenakan MD5, hampir mustahil untuk dipecahkan dengan serangan *brute-force*. Selain itu MD5 tidak seperti proses enkripsi data yang bisa dilakukan proses pembalikan atau dekripsi, yang membuat data bisa terbaca kembali. MD5 juga banyak dipakai untuk mendeteksi file yang terkorupsi, yang bisa disebabkan berbagai macam kemungkinan.

Dalam penelitian ini MD5 akan dicoba untuk memvalidasi dokumen HTML. MD5 merupakan penyempurnaan dari algoritma message digest yang lain yaitu MD2 maupun MD4. Berikut adalah tabel perbandingan MD2, MD4 dan MD5.

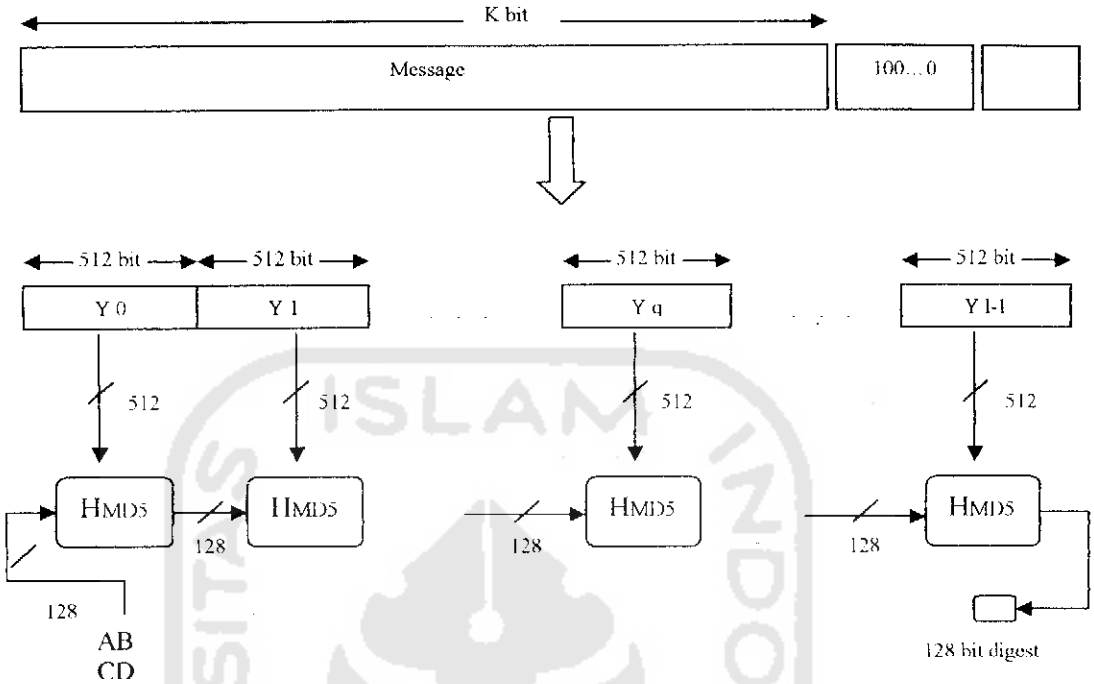
Tabel 2.1 Perbandingan MD2, MD4 dan MD5

	MD2	**MD4**	**MD5**
Waktu pembuatan	1989	1990	1991
Pembuat	Ronald L. Rivest		
Desain mesin	8-bit	32-bit	32-bit
Digest output	128-bit		
Kecepatan proses	Paling lambat	Paling cepat	Cepat
Keamanan	Paling lemah	Kuat	Paling kuat

Dalam perkembangannya, MD5 digunakan untuk pembandingan file, keamanan pengiriman data lewat jaringan, bahkan MD5 digunakan untuk mengetahui status keaslian suatu file. Untuk kegunaan yang terakhir, situs resmi PHP, yaitu *www.php.net,* menggunakan MD5 untuk melindungi keaslian *file-file* php yang disediakan untuk di*download*, sehingga apabila file-file tersebut disisipi program virus atau *trojan*, bisa dideteksi dengan mudah.

2.2.4 Cara Kerja MD5

Cara kerja MD5 dalam mengubah data, berapapun panjangnya, terdiri dari lima langkah. Untuk menghasilkan message digest 128-bit, data input diproses di dalam blok-blok 512-bit [STA95]. Dibawah ini adalah gambar algoritma MD5.



Gambar 2.1 Ilustrasi proses pembuatan MD5

Penambahan Bit Tambahan

Data input ditambahkan bit tambahan sehingga panjang data input, dalam satuan bit, setara dengan 448 mod 512 atau 64 bit. Panjang bit tambahan selalu lebih kecil 64 bit dibandingkan kelipatan 512 bit.

Bit tambahan selalu ditambahkan, meskipun panjang data input sudah memenuhi ketentuan. Contoh, jika panjang masukan adalah 448 bit, maka input tersebut ditambahkan bit tambahan sepanjang 512 bit, sehingga panjang input menjadi (448+512) mod (2*512) atau 64 bit. Panjang bit tambahan berkisar antara 1 sampai 512 dan terdiri atas bit 1 dan bit 0.

Penambahan Panjang

Proses ini menggabungkan data input dengan bit tambahan pada proses pertama. Jika panjang data sebelum diproses pada langkah pertama lebih besar

dari 2^{64}, maka hanya 64 bit terakhir saja yang digunakan. Sehingga panjang data sekarang adalah panjang data input mod 2^{64}.

Data hasil yang diperoleh dari dua langkah pertama adalah data yang panjangnya merupakan kelipatan dari 512 bit. Dari gambar 2.1, data yang telah ditambah panjangnya, digambarkan sebagai rangkaian dari blok-blok 512 bit ($Y_0, Y_1, \ldots, Y_{L-1}$), jadi panjang keseluruhan pesan yang telah melalui dua proses adalah L x 512 bit.

Inisialisasi Buffer MD

Tahap ini adalah tahap pembuatan buffer. Buffer direpresentasikan sebagai empat buah register yang dinamakan A,B,C,D yang masing-masing besarnya 32 bit.

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

Pemrosesan Data dalam Blok 512 bit

Langkah ke empat ini terdiri atas empat bagian pemrosesan. Setiap bagian mempunyai struktur yang mirip, tapi memakai empat fungsi yang berbeda, dalam hal ini dimisalkan F, G, H, I.

Pada proses ini, tiap bagian pemrosesan menjadikan tiap blok 512 bit dan buffer 128 bit sebagai input. Input tersebut digunakan untuk melakukan update terhadap isi buffer.

Output

Setelah semua blok 512 bit selesai diproses, maka keluaran dari blok 512 bit terakhir adalah message digest dari data input

2.2.5 Kekurangan dan Kelebihan MD5

Kelebihan MD5

1. Sulit untuk dipecahkan walaupun dengan serangan brute force.
2. Tingkat keamanan MD5 adalah salah satu yang terbaik.
3. Tidak bisa diubah kembali menjadi data asli *(irreversible)*
4. Hasil keluaran MD5 selalu 32 karakter.

Kekurangan MD5

1. Proses perubahan data asli menjadi MD5 perlu waktu relatif lama.
2. Memerlukan hardware komputer yang cukup besar, semakin besar kemampuan komputer makin cepat kinerjanya.

2.3 HTML

HTML adalah bahasa yang digunakan untuk menulis halaman *web*. HTML merupakan pengembangan dari standar format dokumen teks yaitu *Standard Generalized Markup Language (SGML)*. HTML sebenarnya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada satu sistem operasi tertentu.

2.3.1 Sejarah Singkat HTML

HTML dibuat oleh Tim Berners-Lee dan dipopulerkan pertama kali oleh *browser* Mosaic. Selama awal tahun 1990, HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML selalu akan terdapat penambahan

kemampuan dan fasilitas yang lebih baik dari versi sebelumnya, namun hal itu tidak sampai mengubah cara kerja HTML.

HTML 2.0 secara resmi dikeluarkan pada bulan November 1995 oleh *IETF (Internet Engineering Task Force)*. Menyusul setelah itu HTML 3.0 dengan penambahan fitur yang lebih lengkap dibanding versi 2.0. Kemudian *World Wide Web Consortium's (W3C) HTML Working Group* merilis HTML 3.2 pada Januari 1997. Selanjutnya *W3C* yang kemudian merilis HTML versi berikutnya [PUR01].

2.3.2 Struktur Dokumen HTML

Penulisan HTML dapat dilakukan dengan editor teks seperti *Notepad* yang dimiliki oleh sistem operasi *Windows*, atau dapat juga menggunakan editor lain yang mampu mengolah dan menyimpan file dalam bentuk teks.

HTML terdiri dari tanda-tanda yang dinamakan *tag*. *Tag* selalu ditulis diantara tanda lebih kecil dan tanda lebih besar (*<tag>*).

Secara sederhana, dokumen HTML terdiri dari dua bagian yaitu *header* dan *body*. Struktur HTML diapit oleh tag `<html>` dan `</html>`. Standar penulisannya adalah :

```
<html>
<head>
        Deskripsi dokumen
</head>
<body>
        Isi dokumen
</body>
</html>
```

2.4 VALIDASI DOKUMEN HTML DENGAN ALGORITMA MD5

Validasi dokumen HTML artinya dokumen HTML tersebut akan diberikan sebuah tanda yang menjadi bukti keasliannya yaitu tanda tangan digital. Tanda keaslian tersebut sangat bergantung kepada input, dalam hal ini isi dokumen teks HTML. Tanda keaslian ini juga menjadi patokan dalam pemeriksaan keaslian dokumen. Jika suatu dokumen bernilai valid, maka berarti data tersebut sesuai dengan aslinya atau belum mengalami perubahan. Jika data tersebut tidak valid, maka data itu sudah tidak asli lagi atau telah mengalami perubahan.

Untuk membuat sistem yang mampu menghasilkan tanda tangan digital maupun melakukan pemeriksaan kevalidan suatu dokumen HTML, maka diperlukan algoritma MD5. Dalam sistem ini MD5 berperan untuk membuat tanda tangan digital dari dokumen HTML dan tanda tangan digital tersebut akan disisipkan dalam dokumen yang bersangkutan. Untuk pemeriksaan validasi, dokumen HTML yang diperiksa akan diuji berdasarkan tanda tangan digital yang telah disisipkan sebelumnya.