TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Digital Singnature Pada File Dokumen Menerapkan Fungsi Hash Dengan Metode MD5

Ririn Saragih

Program Studi Teknologi Informatika, Universitas Budi Darma, Medan, Indonesia
Email: ririnsaragih732@gmail.com

Abstrak−Dokumen yang memiliki *digital signature* dimaksud untuk memvalidasi dari mana data tersebut berasal. *Digital dsignature* dapat dilakukan melalui enkripsi. Algoritma yang biasanya dipakai untuk sebuah *digital signature* pada *file* dokumen yaitu algoritma *Message Digest* (MD5) yang juga salah satu fungsi *hash*. Dan perlu dilakukan keaman dan keaslian *Digital signature* pada *file* dokumen agar tidak mudah dimanipulasi oleh orang lain.Metode yang sesuai dengan Permasalahan ini dengan menggunakan algoritma yang biasanya dipakai untuk membuat *digital signature* yaitu algoritma MD5 yang merupakan algoritma *hash* dari jenis MD5 atau arah untuk mendapatkan nilai hash sepanjang 128 bit. Algoritma MD5 dapat digunakan untuk melakukan pengecekan integritas data, pembuatan *digital signature* pada *file* dokumen dan lain-lain. Sebuah *digital signature* dari pengamanan dan keaslian *file* dokumen juga terdapat salah satu fungsi hash pada jurnal yang berjudul impelementasi algoritma MD5 untuk keamanan dokumen.Dengan menggunakan fungsi *hash* satu arah maka akan menghasilkan *message Digest* dari pesan asli. Dikembangkan dari MD5 algoritma yang dikembangkan oleh Ron Rivest, menerima masukan pesan berbagai panjang dan menghasilkan kode hash 128-bit ini telah menjadi salah satu algoritma *hash* yang paling banyak digunakan. Algoritma ini pada dasarnya dibuat untuk keamanan dan keaslian *digital signature* pada *file* dokumen fungsi *hash* tidak dapat menjawab semua masalah yang ada seperti masalah kerahasiaan.

Kata Kunci: Digital Signature; Hash

Abstract−Documents that have a digital signature are intended to validate where the data comes from. Digital signature can be done through encryption. The algorithm that is usually used for a digital signature on a document file is the Message Digest (MD5) algorithm which is also a hash function. And it is necessary to do the security and authenticity of the Digital signature on the document file so that it is not easily manipulated by others. The method that suits this problem is to use an algorithm that is usually used to create digital signatures, namely the MD5 algorithm which is a hash algorithm of the MD5 type or direction to get the value. hash is 128 bits long. The MD5 algorithm can be used to check data integrity, create digital signatures on document files and others. A digital signature of the security and authenticity of the document file there is also a hash function in the journal entitled Implementing the MD5 algorithm for document security. By using a one-way hash function, it will generate a message Digest from the original message. Developed from the MD5 algorithm developed by Ron Rivest, accepting input messages of various lengths and generating a 128-bit hash code it has become one of the most widely used hash algorithms. This algorithm is basically made for the security and authenticity of digital signatures in document files, the hash function cannot answer all existing problems such as confidentiality issues.

Keywords: Digital Signature; Hash

1. PENDAHULUAN

Citra adalah suatu gambar atau kemiripan dari suatu objek citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses oleh komputer secara langsung. Tentu agar bisa diproses didalam komputer, citra analog harus dikonvrsi menjadi citra digital. Citra digital adalah citra yang dapat diolah oleh komputer, sedangkan citra yang dihasilkan dari peralatan digital (citra digital) langsung bisa diolah oleh komputer. Citra merupakan hasil evaluasi dalam diri seseorang berdasarkan persepsi dan pemahaman terhadap gambar yang telah diolah, di organisasikan, dan disimpan dalam benak seseorang, citra dapat diukur melalui pendapat, kesan atau respon seseorang dengan tujuan untuk mengetahui secara pasti apa yang ada dalam pikiran setiap individu mengenai suatu objek, bagaimana mereka memahaminya dan apa yang mereka sukai atau yang tidak disukai dari objek tersebut. Defenisi digital signature pada file dokumen adalah kumpulan dari data dan informasi yang saling.berhubungan file dokumen ini juga tersimpan di ruang penyimpanan sekunder.

Dokumen yang memiliki digital signature dimaksud untuk memvalidasi dari mana data tersebut berasal. Digital dsignature dapat dilakukan melalui enkripsi. Algoritma yang biasanya dipakai untuk sebuah digital signature pada file dokumen yaitu algoritma Message Digest (MD5) yang juga salah satu fungsi hash. Dan perlu dilakukan keaman dan keaslian Digital signature pada file dokumen agar tidak mudah dimanipulasi oleh orang lain.

Metode yang sesuai dengan Permasalahan ini dengan menggunakan algoritma yang biasanya dipakai untuk membuat digital signature yaitu algoritma MD5 yang merupakan algoritma hash dari jenis MD5 atau arah untuk mendapatkan nilai hash sepanjang 128 bit. Algoritma MD5 dapat digunakan untuk melakukan pengecekan integritas data, pembuatan digital signature pada file dokumen dan lain-lain. Sebuah digital signature dari pengamanan dan keaslian file dokumen juga terdapat salah satu fungsi hash pada jurnal yang berjudul impelementasi algoritma MD5 untuk keamanan dokumen.

Dengan menggunakan fungsi hash satu arah maka akan menghasilkan message Digest dari pesan asli. Dikembangkan dari MD5 algoritma yang dikembangkan oleh Ron Rivest, menerima masukan pesan berbagai panjang dan menghasilkan kode hash 128-bit ini telah menjadi salah satu algoritma hash yang paling banyak digunakan.

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Algoritma ini pada dasarnya dibuat untuk keamanan dan keaslian digital signature pada file dokumen fungsi hash tidak dapat menjawab semua masalah yang ada seperti masalah kerahasiaan.

2. METODOLOGI PENELITIAN

2.1 Digital Signature

Digital signature adalah hampir sama dengan cara kerja ”tanda tangan” dokumen biasa, terdapat 2 algoritma pada sistem *digital sigature*, yaitu algoritma *sign* untuk menandatangani sebuah dokumen *M* dan menghasilkan sebuah tanda tangan (*sign*) *P*, dan algoritma *verify* yang mengambilkan nilai *true* bila “tanda tangan” *p* memang milik penandatangan dan untuk dokumen *M*. Kegunaan *digital signature* mirip dengan tanda tangan dalam versi nyata, yaitu memberikan kepastian keasliaan dan persetujuan dokumen oleh si penanda tangan dalam *digital signature*, “tanda tangan” adalah dalam bentuk *digital* yang digunakan untuk mengesahkan sebuah dokumen *digital*[1], [2].

2.2 Fungsi *Hash*

Fungsi *hash* adalah sebuah fungsi yang masukannya adalah sebuah pesan dan keluaran sebuah sidik pesan (*message fingerprint*). Sidik pesan sering juga disebut *mesage digest*. Fungsi *hash* dapat digunakan untuk mewujudkan beberapa layanan keamanan jaringan misalnya untuk keutuhan data dan otentikasi pesan. Fungsi *hash* ini menjelaskan tentang konsep dasar yang dipakai dalam sistem kriptografi dan membahas beberapa fungsi *hash* yang banyak dipakai, yaitu MD5 dan *SHA*[3], [4].

Berikut diuraikan sifat-sifat fungsi *hash* antara lain :

1. Tahap *per-image* (*per-image resistent*): Bila diketahui nilai *hash* h, sulit didapatkan (secara komputasi tidak layak) m dimana h=*hash*(m).
2. Tahap *per-image* kedua (*second per-image resistant*) **:** Bila diketahui *input* m_1, sulit dicari *input* m_2 (tidak sama dengan m_1) yang menyebabkan *hash* (m_1)=*hash* (m_2).
3. Tahap tumbukan (*collision-resistant*) **:** Sulit dicari input yang berbeda, m_1 dan m_2, yang menyebabkan *hash*(m_1) = *hash* (m_2).

Fungsi *hash* ini adalah fungsi yang menerima masukan *string* yang panjangnya sembarang dan mengkonversinya menjadi *string* keluaran yang panjangnya tetap (*fixed*), umumnya berukuran juah lebih kecil dari pada ukuran string semula. Fungsi *hash* dapat menerima masukan *string* apa saja, jika *string* menyatakan pesan (*message*), sembarang pesan *M* berukuran bebas dikompresi oleh fungsi *hash* H melalui persamaan berukut:

$$h=H(M) \qquad (1)$$

Keterangan
M= Pesan dengan panjang sembarang
H= Nilai *hash* (*hash value*) atau pesan ringkas (*message digest*)

Keluaran fungsi *hash* disebut juga nilai hash (*hash value*) atau pesan ringkas (*message digest*). Fungsi *hash* akan mengembalikan *hash value* yang panjangnya jauh lebih pendek dibandingkan dengan panjang string masukan, sebagai contoh pesan yang dicari nilai *hash*–nya hanya memiliki ukuran sebesar 1Mb, maka hash *value* yang dihasilkan hanya 128 bit[5]. Kebanyakan fungsi *hash* yang ada saat ini berupa fungsi *hash* satu arah. Artinya, pesan yang sudah diubah menjadi *message digest* tidak dapat dikembalikan lagi menjadi pesan semula (*irreversible)*. Selain itu fungsi *hash* satu arah mempunyai sifat sebagai berikut:

1. Fungsi H dapat diterapkan pada blok data berukuran beberapa saja.
2. H menghasilkan nilai (h) dengan panjang tetap (*fixed length output*).
3. H (x) mudah dihitung utuk setiap nilai x yang diberikan.
4. Untuk setiap yang h yang dihasilkan tidak mungkin nilai x sedemikian sehingga H(x) =h
5. Untuk setiap x yang diberikan, tidak mungkin dicari y $\neq$ x sedemikian sehingga H(y)= H(x).
6. Tidak mungkin dicari pasangan x dan y sedemikian sehingga H(x) = H(y).

Persamaan fungsi *hash* satu arah dapat dilihat pada persamaan 2

$$hi=H(Mi, hi-1) \qquad (2)$$

Tabel 1. Skema Fungsi *Hash*

T[1]=D76AA478	T[17]=F61E2562	T[33]=FFFA3942	T[49]=F4292244
T[2]=E8C7B756	T[18]=C040B340	T[34]=8771F681	T[50]=432AFF97
T[3]=242070DB	T[19]=265E5A51	T[35]=69D96122	T[51]=AB9423A7
T[4]=C1BDCEEE	T[20]=E9B6C7AA	T[36]=FDE5380C	T[52]=FC93A039
T[5]=F57C0FAF	T[21]=D62F105D	T[37]=A4BEEA44	T[53]=655B59C3
T[6]=4787C62A	T[22]=02441453	T[38]=4BDECFA9	T[54]=8F0CCC92
T[7]=A8304613	T[23]=D8A1E681	T[39]=F6BB4B60	T[55]=FFEFF470

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

2.3 Metode MD5

Metode adalah fungsi dari *hash* satu arah yang dibuat oleh *Ronald Rivest* pada tahun 1991. MD5 merupakan perbaikan dari MD4 berhasil diserang oleh kriptanalis. MD5 dikembangkan dari MD, MD2, MD3 dan MD4, MD5 pesan mencerna algoritma, yang yang dikembangkan oleh *Ron Rivest*, menerima masukan pesan berbagi panjang dan menghasilkan kode *hash* 128-bit. Telah menjadi salah satu algoritma *hash* yang paling banyak digunakan dalam algoritma ini pada dasarnya dirancang untuk tujuan keamaan yang tinggi dimana pesan yang besar harus "kompresi" dengan cara aman sebelum ditanda tangani dengan kunci pribadi. Algoritma MD5 digunakan mengimplentasikan integritas pesan yang menghasilkan *message digest* dari ukuran 128-bit. Dalam implementasi algoritma MD5 menggunakan algoritma 128 bit sebagi unsur dasar dan membuat apilikasi untuk 640 pesan bit sehingga menciptakan keamanan yang tingi untuk transfer data dalam jaringan 3G, 4G, dapat digunakan untuk mengirim f*ile JPG, MPEG, DOCX, PDF*. Ini adalah fungsi matematika yang memproses informasi untuk membuat pesan yang berbeda dan unik. Keuntungan lain adalah bahwa pesan yang dibuat jauh lebih pendek dari dokumen aslinya. Memproses pesan dan menghasilkan 128-bit *message digest.*

Langkah-langkah pembuatan *message digest* secara garis besar adalah menambahkan *padding bits*, menambahkan nilai panjang pesan semula, menginisialisasi penyangga (*buffer*) MD, dan mengolah pesan dalam blok berukuran 512 bit.

1. Menambahkan *padding bits*.
   a. Pesan ditambah dengan sejumlah *padding bits* sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 *modulo* 512.
   b. Jika panjang pesan 448 bit, tambahkan 512 bit sehingga menjadi 960 bit. Jadi, panjang *bit padding bits* adalah antara 1 sampai 512.
   c. *Padding bits* terdiri atas sebuah bit 1 dan, sisanya, yang mengikutinya, bit 0.
2. Menambahkan nilai panjang pesan.
   a. Pesan yang telah diberi *padding bits* selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula.
   b. Jika panjang pesan >2^{64}, yang diambil adalah panjangnya dalam modulo 2^{64}. Dengan kata lain, jika panjang pesan semula adalah K bit, 64 bit yang ditambahkan menyatakan K modula2^{64}.
   c. Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512bit.
3. Menginisialisasi penyangga MD
   a. MD5 membutuhkan 4 buah penyangga yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 4 x 32 = 128 bit. Keempat penyangga ini menampung hasil antara dan hasil akhir.
   b. Keempat penyangga dinamai A, B, C, dan D. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi *HEX*) berikut:
      A = 01 23 45 67
      B = 89 AB CD EF
      C = FE DC BA 98
      D = 76 54 32 10
4. Mengolah pesan dalam blok berukuran 512 bit.
   a. Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y_0sampai Y_{L-1}).
   b. Setiap blok 512 bit diproses bersama dengan penyangga MD menjadi keluaran 128 bit, yang disebut proses H_{MD5} seperti diperlihatkan pada Gambar 2.2. Proses H_{MD5} terdiri atas 4 buah putaran. Masing-masing putaran melakukan operasi dasar memakai sebuah elemen T. Jadi, setiap putaran memakai 16 elemen tabel T.
   c. Pada Gambar 2.2, Yang menyatakan blok 512 bit ke-q dari pesan yang telah ditambahi *padding bits* dan tambahan 64 bit nilai panjang pesan semula. MDq adalah nilai message digest 128 bit dari proses H_{MD5} ke-q. Pada awal proses, MDq berisi nilai inisialisasi penyangga MD.
   d. Operasi dasar MD5 dapat ditulis dalam bentuk persamaan berikut
      a $\leftarrow$ b + CLSs (a + g(b, c, d) + X [k] + T [i])

Keterangan :

a, b, c, d	: empat buah peubah penyangga 32 bit (berisi nilai penyangga A, B, C,D)
g	: salah satu fungsi F, G, H, I
CLSs	: circular *left shift* sebanyak s bit
X [k]	: kelompok 32 bit ke-k dari blok 512 bit *message* ke-q.
Nilai k	: 0 sampai 15
T [i]	: elemen tabel T ke-I (32 bit)
+	: operasi penjumlahan modulo 232

Fungsif_F, f_G, f_H, dan f_I adalah fungsi untuk memanipulasi masukan a, b, c, dan d dengan ukuran 32 bit. Masing-masing fungsi dapat dilihat pada tabel 2.

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Tabel 2. Fungsi-Fungsi Dasar MD5

Nama	Notasi	g (b, c, d)
Ff	F (b, c, d)	(b ∧ c) ∨ (~ b ∧ d)
Fg	G (b, c, d)	(b ∧ d) ∨ (c ∧ ~ d)
Fh	H (b, c, d)	b ⊕ c ⊕ d
Fi	I (b, c, d)	c ⊕ (b ∧ ~ d)

Sebagai catatan, operator logika AND, OR, NOT dan XOR masing-masing dilambangkan dengan ∧,∨,~,⊕. Nilai T[i] dapat dilihat pada tabel 3 yang disusun oleh fungsi $2^{32} \times$ abis (sin(i)) dengan i dalam radian]

Tabel 3. Nilai T [I]

T[1]=D76AA478	T[17]=F61E2562	T[33]=FFFA3942	T[49]=F4292244
T[2]=E8C7B756	T[18]=C040B340	T[34]=8771F681	T[50]=432AFF97
T[3]=242070DB	T[19]=265E5A51	T[35]=69D96122	T[51]=AB9423A7
T[4]=C1BDCEEE	T[20]=E9B6C7AA	T[36]=FDE5380C	T[52]=FC93A039
T[5]=F57C0FAF	T[21]=D62F105D	T[37]=A4BEEA44	T[53]=655B59C3
T[6]=4787C62A	T[22]=02441453	T[38]=4BDECFA9	T[54]=8F0CCC92
T[7]=A8304613	T[23]=D8A1E681	T[39]=F6BB4B60	T[55]=FFEFF47D
T[8]=FD469501	T[24]=E7D3FBCB	T[40]=BEBFBC70	T[56]=85845DD1
T[9]=698098D8	T[25]=21E1CDE6	T[41]=289B7EC6	T[57]=6FA87E4F
T[10]=8B44F7AF	T[26]=C33707D6	T[42]=EAA127FA	T[58]=FE2CE6E0
T[11]=FFFF5BBI	T[27]=F4D50D87	T[43]=D4EF3085	T[59]=A3014314
T[12]=895CD7BE	T[28]=455A14ED	T[44]=0488D051	T[60]=4E0811A1
T[13]=6B901122	T[29]=A9E3E905	T[45]=D9D4D039	T[61]=F7537E82
T[14]=FD987193	T[30]=FCEFA3F8	T[46]=E6DB99E5	T[62]=BD3AF235
T[15]=A679438E	T[31]=676F02D9	T[47]=1FA27CF8	T[63]=2AD7D2BB
T[16]=49B40821	T[32]=8D2A4C8A	T[48]=C4AC5665	T[64]=EB86D391

Setelah putaran keempat, a, b, c, dan d ditambahkan ke A, B, C, dan D. Selanjutnya, algoritma memproses blok data berikutnya(Y_{q+1}). Ketentuan akhir dari algoritma MD5 adalah hasil penyambungan bit di A, B, C, dan D. Dari uraian di atas, secara umum fungsi *hash* MD5 dapat ditulis dalam persamaan matematis berikut:

$$MD_0=\text{IV}$$

$$MD_{Q+1}=\text{MDq+fi(Yq+fH(YQ+Ff(Yq+MDq))))}$$

$$\text{MD=MDL-1}$$

Keterangan
IV : intinal vector dari penyangga ABCD, yang dilakukan pada proses inisialisasi penyangga
Yq : blok pesan berukuran 512-bit ke-q
L : jumlah blok pesan
MD : nilai akhir *message digest*

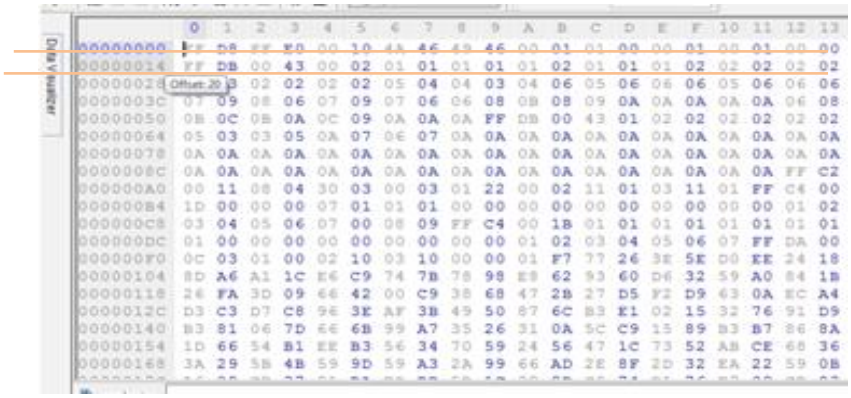
3. HASIL DAN PEMBAHASAN

Keamanan dan keaslian *digital segnature* pada *file* dokumen merupakan salah satu aspek terpenting yang harus diperhatiakan dan diamankan. Karena sering terjadi pemalsuan *digital signature* pada *file* dokumen yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab *file* dokumen merupakan barang bukti *digital signature* yang salah satunya berasal dari *Handphone*, dalam hal kejahatan digital signature pada *file* dokumen biasanya dimanipulasi untuk menghasilkan bukti-bukti yang ada didalamnya, oleh sebab itu diperlukan analisis forensik untuk dapat mengetahui *file* dokumen tersebut. Ada pun berubahan *file* dokumen yang merubah bentuk aslinya adalah berupa JPG

Metode ini merupakan algoritma *hash* baik digunakan untuk algoritma membuat suatu *digital signature* pada *file* dokumen dengan metode MD5. *Digital signature* pada *file* dokumen merupakan salah satu aspek terpenting yang harus diperhatikan dan diamankan. Dalam hal ini, sangat terkait hubungannya dengan dokumen-dokumen penting agar tidak mudah ditiru oleh orang lain dan disalah gunakan untuk kepentingan tertentu.

Sangat penting menjaga keamanan dan keaslian *digital signature* pada *file* dokumen Analisa masalah bertujuan untuk melakukan keamanan persoalan-persoalan yang muncul di sistem, hal ini dilakukan agar suatu proses tidak terjadi kesalahan-kesalahan. Dalam analisa masalah ini, masalah yang akan dianalisa yaitu keamanan *digital signature* pada *file* dokumen.

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin



Gambar 1. Data byte Digital signature

Dari data tersebut diambil sebanyak 20 *byte* atau 40 karakter heksadesimal dan konversikan ke biner, berguna untuk mengetahui nilai biner bilangan tersebut.

Dari gambar data dokumen di atas diambil sebanyak 20 *byte* untuk plaintexs, yaitu: FFD8FFE000104A4649460001010000010010000

3.1 Penerapan Algoritma MD5

Metode MD5 menerima masukan pesan berbagai panjang dan menghasilkan kode hash128-bit telah menjadi salah satu algritma hash yang paling banyak digunakan dalam algritma dasarnya bertujuan untuk keamanan yang tinggi yang mana pesan harus 'kompres' dengan cara aman. Algritma MD5 digunakan mengimplementasikan intergritas pesan yang menghasilkan message Digest dari ukuran 128-bit. Impelementasi algritma MD5 menggunakan algritma 128 bit unsur dasar membuat pesan. Lalu memproses pesan dan menghasilkan 128-bit *messeage digest*

3.2 Penambahan Bit-bit Pengganjal Dan Nilai Panjang Pesan Semulai

Dari data yang digunakan sebagai plaintexs diubah ke biner.
11111111 11011000 11111111 11100000 00000000 00000001 01001010
01000110 01001001 01000110 00000000 00000000 00000001 00000001 00000000 00000000 00000001 00000000 00000000
Data sebanyak 20 *byte* di atas diketahui 160 bit, untuk mencukupi 512 bit ditambah bit-bit pengganjal (padding bits) sebanyak 344 dan 8 bit jumlah pesan semua, kerana pada MD5 memproses blok-blok bit yang berjumlah 64 blok atau 512 bit. Bit pengganjai yang ditambahi dimulai bit 1 diikuti bit 0 selebihnya sehingga urutan bit 464. Untuk 8 bit terakhir menyatakan jumlah karakter dalam notasi biner yaitu 160 = 11100000

Berikut ini adalah urutan blok-blok bit setelah ditambahkan :.
11111111 11011000 11111111 11100000 00000000 00000001 01001010
01000110 01001001 01000110 00000000 00000000 00000001 00000001 00000000
00000001 00000000 00000001 00000000 00000000
10000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 **11100000**

Bit yang bercetak tebal adalah bit dari plaintexs *file* berbntuk dokumen bit 1 pada urutan ke 160 yang bercetak tebal adalah awal dari bit pengganjal dan dikuti bit-bit 0 dan 8 bit terakhir bercetak tebal adalah bit yang menyatakan jumlah 160

3.3 Menginisalisasi Penyangga (MD)

MD5 membutuhkan 4 buah Penyangga yang masing-masing panjangnya 32 bit. Total panjang Penyangga adalah 4x32 = 128 bit. Keempat Penyangga ini menampung hasil antara hasil dan akhir.
Keempat Penyangga dinamai A, B, C, dan D. Setiap Penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) berikut:

A = 01 23 45 67
B = 89 AB CD EF
C = FE DC BA 98
D =76 54 32 10

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Fungsif_F, f_G, f_H, dan f_I adalah fungsi untuk memanipulasi masukan a, b, c, dan d dengan ukuran 32 bit. Masing-masing fungsi dapat dilihat pada tabel 4.

Tabel 4. Fungsi-Fungsi Dasar MD5

Nama	Notasi	g (b, c, d)
Ff	F (b, c, d)	(b ∧ c) ∨ (~ b ∧ d)
Fg	G (b, c, d)	(b ∧ d) ∨ (c ∧ ~ d)
Fh	H (b, c, d)	b ⊕ c ⊕ d
Fi	I (b, c, d)	c ⊕ (b ∧ ~ d)

Penyelesaian fungsi-fungsi MD5

1. F (b,c,d) = (b ∧ d) ∨ (~ b ∧ d)
   = (89 AB CD EF ∧ FE DC BA 98) ∨ (~89 AB CD EF ∧ 76543210)
   a. (b ∧ c) = (89 AB CD EF ∧ FE DC BA 98)
      = 1000 1001 1010 1011 1100 1101 1110 1111 ∧
      1111 1110 1101 1100 1011 1010 1001 1000
      = 1000 1000 1000 1000 1000 1000 1000 1000
   b. (~ b ∧ d) = (~89 AB CD EF ∧ 76543210)
      = ~ 0111 0110 0101 0100 0011 0010 0001 0000 ∧
      0111 0110 0101 0100 0011 0010 0001 0000
      = 111 0110 0101 0100 0011 0010 0001 0000
   c. (b ∧ d) ∨ (~ b ∧ d)
      = (89 AB CD EF ∧ FE DC BA 98) ∨ (~89 AB CD EF ∧ 76543210)
      = 1000 1001 1010 1011 1100 1001 1110 1111 ∨
      0111 0110 0101 0100 0011 0010 0001 0000
      = 1111 1110 1101 1100 1011 1010 1001 1000
      F E D C B A 9 8
2. G (b,c,d) = (b ∧ d) ∨ (~ c ∧ d)
   = (89 AB CD EF ∧ FE DC BA 98) ∨ (~89 AB CD EF ∧ 76543210)
   a. (~ b ∧ d) = (89 AB CD EF ∧ 76543210)
      = 1000 1001 1001 1011 1100 1001 1110 1111 ∧
      0111 0110 0101 0100 0011 0010 0001 0000
      = 0000 0000 0000 0000 0000 0000 0000 0000
   b. (c ∧~ d) = (FE DC BA 98 ∧ ~ 76543210)
      = 1111 1110 1101 1100 1011 1010 1001 1000 ∧
      ~ 1000 1001 1010 1011 1100 1101 1110 111
      = 1000 1000 1000 1000 1000 1000 1000 1000
   c. (c ∧ d) ∨ (c ∧~ d)
      = 0000 0000 0000 0000 0000 0000 0000 0000 ∨
      1000 1000 1000 1000 1000 1000 1000 1000
      = 1000 1000 1000 1000 1000 1000 1000 1000
      8 8 8 8 8 8 8 8
3. H (b.c.d) = b ⊕ c ⊕ d
   = 89 AB CD EF ⊕ FE DC BA 98 ⊕ 76543210
   = 1000 1001 1010 1011 1100 1101 1110 1111
   1111 1110 1001 1100 1011 1010 1001 1000
   0111 0110 0101 0100 0011 0010 0001 0000 ⊕
   = 0000 0001 0010 0011 0100 0101 0110 0111
   0 1 2 3 4 5 6 7
4. I (b,c,d) = c ⊕(b ∧~ d)
   C = 1111 1110 1101 1100 1011 1010 1001 1000
   (b ∧~ d) = (89 AB CD EF ∧ ~ 76543210)
   = 1110 1001 1010 1011 1100 1101 1110 1111
   ~ 1110 1001 1010 1011 1100 1101 1110 1111
   1110 1001 1010 1011 1100 1101 1110 1111 ∧
   c⊕ (b ∧~d) = 1111 1110 1101 1100 1011 1010 1001 1000
   1000 1001 1010 1011 1100 1101 1110 1111 ⊕
   = 0111 0111 0111 0111 0111 0111 0111 0111
   7 7 7 7 7 7 7 7

   Inisialisasi penyangga dalam biner, yaitu:
   A= 0 1 2 3 4 5 6 7

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

= 0000 0001 0010 0011 0100 0101 0110 0111
B= 8 9 A B C D E F
= 1000 1001 1010 1011 1100 1101 1110 1111
C= F E D C B A 9 8
=1111 1110 1101 1100 1011 1010 1001 1000
D= 7 6 5 4 3 2 1 0
= 0111 0110 0110 0100 0011 0010 0001 0000

3.4 Pengolahan Pesan Dalam Blok Berukuran 512 Bit (Parsing)

Semua bit plaintexs yang berjumlah 512 bit dibagi 16 blok yang mana setiap blok berisi 32 bit bagian berikut adalah 16 blok bit tersebut:

Pesan
M_0 = 11111111 11011000 11111111 11100000
M_1 = 00000000 00000001 01001010 01000110
M_2 = 01001001 01000110 00000000 00000000
M_3 = 00000001 00000001 00000000 00000000
M_4 = 00000000 00000001 00000000 00000000
M_5 = 10000000 00000000 00000000 00000000
M_6 = 00000000 00000000 00000000 00000000
M_5 = 00000000 00000000 00000000 00000000
M_7 = 00000000 00000000 00000000 00000000
M_8 = 00000000 00000000 00000000 00000000
M_9 = 00000000 00000000 00000000 00000000
M_{10} = 00000000 00000000 00000000 00000000
M_{11} = 00000000 00000000 00000000 00000000
M_{12} = 00000000 00000000 00000000 00000000
M_{13} = 00000000 00000000 00000000 00000000
M_{14} = 00000000 00000000 00000000 00000000
M_{15} = 00000000 00000000 00000000 **11100000**

1. Putaran 1:16 operasi dasar dengan g(b,c,d) = f(b,c.d)
   Putaran 1
   A⟶89 AB CD EF((01234567+ FE DC BA 98+504B0304140+D76AA478
   Data biner
   01234567 = 0000 0001 0010 0011 0100 0101 0110 0111
   FE DC BA98 = 1111 1110 1101 1100 1011 1010 1001 1000
   M_0 = 0001 0000 0100 1011 0000 0011 0000 0100
   D76AA478 = 1101 0111 0110 1010 1010 0100 1111 1000 +
   0110 0111 1011 0011 1010 0111 0111 1011
   0110 0111 1011 0011 1010 0111 0111 1011 <<< 7
   Tulisan yang bercetak tebal dari kiri digeser kekanan
   1101 1001 1101 0011 1011 1101 1011 0011
   89ABCDEF = 1000 1001 1010 1011 1100 1101 1110 1111 +
   Hasil = 0110 1011 1111 1111 0000 1011 1001 0010
   6 B F F 0 B 9 2
2. Putaran 2
   A⟶89 AB CD EF((01234567+ FE DC BA 98+504B0304140 + E8C7B756
   Data biner
   01234567 = 0000 0001 0010 0011 0100 0101 0110 0111
   FE DC BA98 = 1111 1110 1101 1100 1011 1010 1001 1000
   M_1 = 0001 0100 0000 0000 0000 0110 0000 0000
   E8C7B756 = 1110 1000 1100 0111 1011 0111 0101 0110 +
   1101 1110 1100 1011 1101 0101 0101 0101
   1101 1110 1100 1011 1101 0101 0101 0101 <<<12
   Tulisan yang bercetak tebal dari kiri digeser kekanan
   1011 1101 0101 0101 0101 1101 1110 1100
   89ABCDEF = 1000 1001 1010 1011 1100 1101 1110 1111 +
   Hasil = 0010 1101 0001 1100 1111 1101 1100 1111
   3 D 1 C F D C F
   Putaran 62
   A⟶89 AB CD EF((01234567+ 77777777+504B0304140 +BD34F235
   Data biner
   01234567 = 0000 0001 0010 0011 0100 0101 0110 0111

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

77777777 = 0111 0111 0111 0111 0111 0111 0111 0111
M_{13} = 0000 0000 0000 0000 0000 0000 0000 0000
BD34F23 = 1011 1101 0011 0100 1110 0010 0011 0101
0001 0111 1110 1101 1011 1111 1110 0011
0001 0111 1110 1101 1011 1111 1110 0011<<<10
Tulisan yang bercetak tebal dari kiri digeser kekanan
1011 0110 1111 1111 1000 1100 0101 1111
89ABCDEF = 1000 1001 1010 1011 1100 1101 1110 1111 +
Hasil = 0100 0000 1010 1101 0101 1010 0100 1110
4 0 A D 5 A 4 E
Putaran 63
A⟶89 AB CD EF((01234567+ 77777777+504B0304140 +2AD7D2BB
Data biner
01234567 = 0000 0001 0010 0011 0100 0101 0110 0111
77777777 = 0111 0111 0111 0111 0111 0111 0111 0111
M_{13} = 0000 0000 0000 0000 0000 0000 0000 0000
2AD7D2BB = 0010 1010 1101 0111 1101 0010 1011 1011
1010 1101 0100 1010 1000 1111 1001 10001
1010 1101 0100 1010 1000 1111 1001 10001<<<15
Tulisan yang bercetak tebal dari kiri digeser kekanan
0100 0111 1100 1100 1101 1011 0101 0010
89ABCDEF = 1000 1001 1010 1011 1100 1101 1110 1111 +
Hasil = 1101 0001 0111 1000 1010 0100 0100 0001
D 1 7 8 A 4 4 1

Tabel 5. hasil penyanga bit di A,B,C dan D

Int	A	B	C	D
t_0	01234567	89 AB CD EF	FE DC BA 98	76543210
t_1	E375B981	01234567	89 AB CD EF	FE DC BA 98
t_2	99CB7E7D	E375B981	01234567	89 AB CD EF
t_3	6B60A7BB	99CB7E7D	E375B981	01234567
t_4	445C10A2	6B60A7BB	99CB7E7D	E375B981
t_5	6830AE69	445C10A2	6B60A7BB	99CB7E7D
t_6	06DBFD67	6830AE69	445C10A2	6B60A7BB
t_7	15D91DAC	06DBFD67	6830AE69	445C10A2
t_8	32DBDFE3	15D91DAC	06DBFD67	6830AE69
t_9	49F9399A	32DBDFE3	15D91DAC	06DBFD67
t_{10}	D936B593	49F9399A	32DBDFE3	15D91DAC
t_{11}	E9ADCDEE	D936B593	49F9399A	32DBDFE3
t_{12}	7CFB2DC2	E9ADCDEE	D936B593	49F9399A
t_{13}	09FCF019	7CFB2DC2	E9ADCDEE	D936B593
t_{14}	221DADEC	09FCF019	7CFB2DC2	E9ADCDEE
t_{15}	21E4A426	221DADEC	09FCF019	7CFB2DC2
t_{16}	38AE93	21E4A426	221DADEC	09FCF019
t_{17}	6A576A10	38AE93	21E4A426	221DADEC
t_{18}	62AE2C12	6A576A10	38AE93	21E4A426
t_{19}	CF77F2A5	62AE2C12	6A576A10	38AE93
t_{20}	6A11D463	CF77F2A5	62AE2C12	6A576A10
t_{21}	8BD3D79B	6A11D463	CF77F2A5	62AE2C12
t_{22}	6A6E5DE2	8BD3D79B	6A11D463	CF77F2A5
t_{23}	DFB1D298	6A6E5DE2	8BD3D79B	6A11D463
t_{24}	4456E5D3	DFB1D298	6A6E5DE2	8BD3D79B
t_{25}	FB7E436F	4456E5D3	DFB1D298	6A6E5DE2
t_{26}	7EF76E35	FB7E436F	4456E5D3	DFB1D298
t_{27}	C0458D87	7EF76E35	FB7E436F	4456E5D3
t_{28}	9078BE1D	C0458D87	7EF76E35	FB7E436F
t_{29}	7B626C7C	9078BE1D	C0458D87	7EF76E35
t_{30}	39EFB3FB	7B626C7C	9078BE1D	C0458D87
t_{31}	3DDE09BD	39EFB3FB	7B626C7C	9078BE1D
t_{32}	3F383B40	3DDE09BD	39EFB3FB	7B626C7C
t_{33}	AC7747D5	3F383B40	3DDE09BD	39EFB3FB

TIN: Terapan Informatika Nusantara
Vol 2, No 1, Juni 2021, Hal 6-15
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

t_{34}	45744EED	AC7747D5	3F383B40	3DDE09BD
t_{35}	74584102	45744EED	AC7747D5	3F383B40
t_{36}	B12CD4CA	74584102	45744EED	AC7747D5
t_{37}	B0B1DD1A	B12CD4CA	74584102	45744EED
t_{38}	12FF0A5F	B0B1DD1A	B12CD4CA	74584102
t_{39}	5A9AC4EE	12FF0A5F	B0B1DD1A	B12CD4CA
t_{40}	290B2E8A	5A9AC4EE	12FF0A5F	B0B1DD1A
t_{41}	3107B650	290B2E8A	5A9AC4EE	12FF0A5F
t_{42}	C53FFCDC	3107B650	290B2E8A	5A9AC4EE
t_{43}	28FB8512	C53FFCDC	3107B650	290B2E8A
t_{44}	752F3594	28FB8512	C53FFCDC	3107B650
t_{45}	0221C46C	752F3594	28FB8512	C53FFCDC
t_{46}	6C8A6D60	0221C46C	752F3594	28FB8512
t_{47}	55B26C77	6C8A6D60	0221C46C	752F3594
t_{48}	9393DF7F	55B26C77	6C8A6D60	0221C46C
t_{49}	AFE337F6	9393DF7F	55B26C77	6C8A6D60
t_{50}	1EDDA3C5	AFE337F6	9393DF7F	55B26C77
t_{51}	F1EE6406	1EDDA3C5	AFE337F6	9393DF7F
t_{52}	A4745228	F1EE6406	1EDDA3C5	AFE337F6
t_{53}	596DF767	A4745228	F1EE6406	1EDDA3C5
t_{54}	27F0361E	596DF767	A4745228	F1EE6406
t_{55}	25558B03	27F0361E	596DF767	A4745228
t_{56}	492B39D0	25558B03	27F0361E	596DF767
t_{57}	9E124625	492B39D0	25558B03	27F0361E
t_{58}	A75F07DC	9E124625	492B39D0	25558B03
t_{59}	A922D355	A75F07DC	9E124625	492B39D0
t_{60}	5B86A818	A922D355	A75F07DC	9E124625
t_{61}	842CA58A	5B86A818	A922D355	A75F07DC
t_{62}	40AB5A4E	842CA58A	5B86A818	A922D355
t_{63}	D178A841	40AB5A4E	842CA58A	5B86A818

Setelah putaran ke 63 a,b,c dan d ditambahkan ke A,B,C, dan D
t_{63} D178A841 40 AB5AE 842CA584 5B86A818
A,B,C,D 01234567 89ABCDEF FEDCBA89 76543210
Hasil akhirnya 4428A323, 36A1E85B, E8F6C9CC, FB758FEC

Tabel 6. Hasil nilai MD5

4428A323
36A1E85B
E8F6C9CC
FB758FEC

Berdasarkan dari perhitungan di atas diperoleh nilai MD5 berbentuk bilangan Hexadesimal dari 20 byte 40 karakter yaitu:" 4428A323, 36A1E85B, E8F6C9CC, FB758FEC"

Output Aplikasi *Hasher pro*

Hasil Analisa : **"**4428A032 36A1E85B E8F6C9CC FB758FEC"
Hasil Pengujian Aplikasi *Hasher pro "*4428A032 36A1E85B E8F6C9CC FB758EC"

4. KESIMPULAN

Dari penelitian dapat disimpulkan dengan menerapkan metode MD5 untuk keamanan *digital signature* pada *file* dokumen dengan fungsi *hash.* Keamanan *digital signature* pada *file* dokumen berjuan untuk mengamakan *digital signature* dari suatu dokumen, dan dari orang-orang yang tidak bertanjung jawab atau tidak berhak dalam dokumen tersebut dengan menggunakan metode MD5. Dengan proses pengujian aplikasi *Hasher Pro* pada keamanan *digital signature* pada *file* dokumen. Metode MD5 salah satu proses perhitungan nilai-nilai biner dari pada *file* dokumen 20 *byte* 40 karekte

REFERENCES

[1] D. P. Precilia and A. Izzuddin, "Aplikasi Tanda Tangan Digital (Digital Signature) Menggunakan Algoritma Message Digest 5

(MD5)," *Energy*, vol. 5, no. 1, pp. 14–19, 2016.
[2] O. K. Sulaiman, M. Ihwani, and S. F. Rizki, "MODEL KEAMANAN INFORMASI BERBASIS TANDA TANGAN DIGITAL DENGAN DATA ENCRYPTION STANDARD (DES) ALGORITHM," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 1, pp. 14–19, Sep. 2016.
[3] Endelina, "Implementasi Digital Signature Pada File Audio Menerapkan Metode SHA-256," *J. Informatics Manag. Inf. Technol.*, vol. 1, no. 2, pp. 60–67, 2021.
[4] D. L. Toruan and R. K. Hondro, "Implementasi Metode Secure Hash Algorithm-1 Untuk Mendeteksi Keaslian File Dokumen," *Bull. Comput. Sci. Res.*, vol. 1, no. 2, pp. 48–56, 2021.
[5] S. Agarwal, A. Rungta, R. Padmavathy, M. Shankar, and N. Rajan, "An improved fast and secure hash algorithm," *J. Inf. Process. Syst.*, vol. 8, no. 1, pp. 119–132, 2012.