JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

Analisa Keamanan Tanda Tangan Digital Dengan Menerapkan Metode AHA-1

Janisah, Lince Tomoria Sianturi, Candra Frenki Sianturi, Murdani

Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: manikjanysah@gmail.com
Submitted **14-02-2021**; Accepted **24-06-2021**; Published **29-06-2021**

Abstrak

Pentingnya nilai informasi menyebabkan seringkali informasi yang diinginkan oleh orang tertentu kemudian dapat memodifikasi tersebut. Untuk dapat mengesahkan informasi yang didapatkan, maka harus dilakukan authentication pada informasi sehingga diketahui keaslian informasi. Tanda tangan merupakan penanda atau identitas yang ada pada suatu dokumen. Tanda tangan mempunyai peranan penting dalam memverifikasi dan melegalisasi dokumen. Untuk mengamankan tanda tangan digital dapat diatasi dengan proses fungsi hash, hash yang terdiri dari Secure Hash Algorithm. Metode Secure Hash Algorithm (SHA-1) adalah fungsi hash yang bekerja satu arah, ini berarti pesan yang sudah diubah menjadi message digest tidak dapat dikembalikan menjadi pesan semula. Dua pesan yang berbeda akan selalu menghasilkan nilai hash yang berbeda pula, dan hasil dari input panjang string yang berbeda akan menghasilkan autput dengan panjang string tetap 160 bit.

Kata Kunci: Keamanan; Tanda Tangan Digital; SHA-1

Abstract

The importance of information value causes often the information desired by certain people can then modify it. To be able to validate the information obtained, authentication must be done so that the authenticity of the information is known. Signature is a marker or identity in a document. Signatures have an important role in verifying and legalizing documents. To secure digital signatures, a hash function can be processed, a hash consisting of a Secure Hash Algorithm. The Secure Hash Algorithm (SHA-1) method is a hash function that works in one direction, this means that messages that have been converted into message digest cannot be returned to the original message. Two different messages will always produce different hash values, and results from different string length input will result in an autput with a fixed string length of 160 bits.

Keywords: Security; Digital Signature; SHA-1

1. PENDAHULUAN

Keamanan tanda tangan digital merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi pada suatu dokumen yang dianggap sangat penting, dimana tidak boleh dimanfaatkan oleh pihak lain yang tidak mempunyai wewenang dan tanggungjawab. Pada saat sekarang ini dalam perkembangan teknologi hampir semua dokumen yang kita miliki tersimpan dalam bentuk *softcopy* yang disimpan pada tempat penyimpanan yang kita miliki, baik itu penyimpanan secara *online* atau *offline*. Salah satu bentuk sistem keamanan tersebut adalah dengan menerapkan sistem pengenalan identitas. Salah satu tanda identitas adalah menggunakan sistem biometrika. Terdapat dua jenis biometrika yaitu biometrika fisik dan perilaku, salah satu bentuk biometrik perilaku adalah tanda tangan. Sistem biometrika dapat melakukan dua tugas, yaitu verifikasi dan identifikasi. Verifikasi tangan berarti memeriksa apakah tanda tangan tersebut adalah milik orang yang sudah tedaftar atau tidak, serta memeriksa apakah tanda tangan tersebut asli atau palsu. Sedagkan identifikasi merupakan tanda identitas diri atau penetapan indentitas seseorang. Dalam hal ini identifikasinya menggunakan tanda tangan, tanda tangan merupakan sesutau yang lazim untuk menunjukkan identifikasi atau tanda identitas diri seseorang. Tanda tangan merupakan salah satu media yang digunakan sebagai media identitas pada seseorang untuk keperluan verifikasi dan legalisasi terhadap suatu informasi. Tanda tangan mempunyai peranan penting dalam memverifikasi dan melegalisasi suatu dokumen[1].

Untuk mengetahui cara mengamankan suatu tanda tangan digital dapat dilakukan dengan cara pembuatan pola atau kunci, agar seseorang tidak akan bisa meniru tanda tangan yang ada pada pesan/dokumen yang kita kirim, disini kita memerlukan sistem verifikasi untuk mencegah pembobolan dokumen elektronik oleh oknum yang tidak bertanggung jawab[1]. Tanda tangan adalah jenis objek tulisan tangan yang artistik , sebagai media yang penting untuk menunjukkan keabsahan suatu informasi untuk menunjukkan keabsahan suatu informasi tertulis, maka perlu dilakukan pengecekan atau verifikasi terhadap tanda tangan tersebut, apakah tanda tangan tersebut benar-benar ditulis oleh orang yang bersangkutan, atau tanda tangan tersebut dipalsukan oleh orang lain.

Metode yang sesuai dengan permasalahan ini diatas dengan menggunakan metode SHA-1, fungsi SHA-1 ini merupakan fungsi Hash yang sudah lumayan tua, berdasarkan penelitian sebelumnya pada jurnal yang berjudul Implementasi Secure Hash Algoritma-1 untuk pengamanan data dalam library Pada Pemograman Java, SHA adalah fungsi Hash satu arah yang dibuat oleh NIST(*National Institude Of Standard and Technology*)[2]. SHA dinyatakan sebagai standar fungsi Hash satu arah. SHA dapat dianggap sebagai kelanjutan pendahuluan MD5 dan dapat dilakukan *string* yang berkoresponden dengan *message digest* yang diberikan. SHA-1 adalah algoritma Hash yang paling banyak digunakan publik. SHA-1 merupakan salah satu jenis dari fungsi Hash satu arah. SHA-1 menerima masukan berapa pesan dengan ukuran maksimum 2^{64} bit(2.147.483.648 *gigabyte*) dan menghasilkan nilai Hash dengan panjang 160 bi.

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian *modern* kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berusan dengan keamanan informasi seperti kerahasian, keutuhan data dan otentikasi entitas. Berikut ini adalah rangkuman beberapa mekanisme yang berkembang pada kriptografi *modern*:

1. Fungsi *Hash* adalah fungsi yang melakukan pemetaan pesan dengan panjang sembarang ke sebuah teks khusus yang disebut *message digest* dengan panjang tetap. Fungsi *Hash* umumya dipakai sebagi nilai uji *(check value)* pada mekanisme keutuhan data.
2. Penyandian dengan kunci simetrik *(Symmetric key encipherment)*. Penyandian dengan kunci simetrik adalah penyandianyang kunci enkripsi dan kunci dekripsi bernilai sama. Kunci pada penyandian simetrik diasumsikan bersifat rahasia hanya pihak yang melakukan enkripsi dan dekripsi yang mengetahui nilainya. Oleh karena itu penyandian dengan kunci simetrik disebut juga penyandian dengan kunci rahasia *secret key encipherment.*
3. Penyandian dengan kunci asimetrik *(Asymmetric key encipherment).* Penyandian dengan kunci asimetrik atau sering juga disebut dengan penyandian kunci publik *(public key)* adalah penyandian dengan kunci enkripsi dan dekripsi berbeda nilai. Kunci enkripsi yang juga disebut dengan kunci publik *(public key)* bersifat terbuka. Sedangkan, kunci dekripsi yang disebut kunci privat *(private key)* bersifat tertutup / rahasia[2].

2.2 Tanda Tangan Digital

Tanda tangan digital atau digital signature merupakan suatu tanda tangan (penanda) yang dibubuhkan pada data digital. Tanda tangan digital bukan merupakan hasil pemindaian atau input tanda tangan melalui antarmuka tertentu. Tanda tangan digital adalah suatu nilai kriptografi yang bergantung pada isi data itu sendiri serta kunci yang digunakan untuk membangkitkan nilai kriptografinya sehingga nilai setiap tanda tangan digital dapat selalu berbeda tergantung pada data yang ditanda tangani.[3].

2.3 Daging Ayam

SHA adalah fungsi has satu arah yang dibuat oleh NIST dan digunakan bersama DSS (*Digital Signature Standard).* Oleh NSA, SHA dinyatakan sebgai standard fungssi hash satu arah. SHA dapat dianggap sebagai kelanjutan pendahuluannya, MD5, yang telah diguankan secara luas. SHA disebut aman *(secure)* karena ia diracang sedemikian rupa sehinggasecara komputasi tidak mungkin menemukan pesan yang berkoresponden dengan *message digest* yang diberikan. SHA merupakan keluarga fungsi Hash satu-arah. Fungsi Hash SHA yang paling umum digunakan adalah SHA-1 yang telah diimplementasikan di dalam berbagai aplikasi dan protokol keamanan seperti TLS, SSl, PGP, SSH, S/MIME, dan *Ipsec*. Anggota pertama keluarga SHA adalah SHA-0 sering diacu sebagai SHA saja. Berikiutnya pada tahun1995 SHA-1 dipublikasikan. Empat varian lain juga telah dipublikasikan yaiyu SHA-224, SHA-256, SHA-384, dan SHA-512. Keempat varian ini diaau seabgai SHA-2. Upa-bab ini hanya menjelaskan Algoritma SHA-1 saja, sedangjan varian lainya tidak dibahas.SHA-1 adalah menerima masukan berupa pesan dengan ukuran maksimal 2^{64} bit (2.147.483.684 *gigabyte)* dan menghasilkan *message digest* yang panjangnya 160 bit, lebihpanjang dari *message digest* yang dihasilkan oleh MD5 yang hanya 128 bit. Gambaran umum pembuatan *message digest* dengan algoritma SHA-1[4].

Adapun teori-teori operasi pada kerja SHA-1 adalah sebagai berikut :

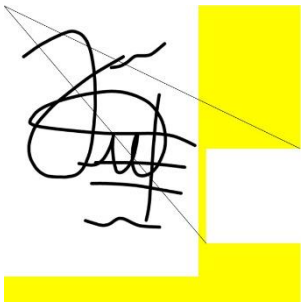
1. Rotate Left (ROL) merupakan operasi bit dimana bit-bit datanya akan diputar kekiri sebanyak operan2 kali, operasi ini hampir sama dengan SHL, namun ROL tetap mempertahankan bit asli datanya.
2. Shift Left (<<) adalah sebuah orator yang digunakan untuk melakukan pergeseran bit ke arah kiri sebanyak nilai yang didefenisikan, operator << akan mengalikan suatu nilai dengan 2.
3. XOR merupakan operasi logika dimana keluaran akan bernilai 1 jika kedua masukannya berbeda nilainya dan bernilai 0 jika kedua masukannya sama nilainya.
4. AND merupakan operasi logika dimana keluaran akan bernilai 1 jika kedua masukannya bernilai 1 juga.

Pesan diberi tambahan untuk membuat panjangnya menjadi kelipatan 512 bit (1 x 512). Jumlah bit asal adalah k bit.Tambahkan bit secukupnya sampai 64 bit kurangnya dari kelipatan 512 (512 – 64 =448), yang disebut juga *kongruen* dengan 448 (mod 512). Kemudian tambahkan 64 bit yang menyatakan panjang pesan. Inisiasi 5 md variable dengan panjang 32 bit yaitu a,b,c,d,e. Pesan dibagi menjadi blok-blok berukuran 512 bit dan setiap blok diolah. Kemudian keluaran setiap blok berikutnya, sehingga diperoleh *output (diggest)*[5].

3. HASIL DAN PEMBAHASAN

Keamanan tanda tangan digital untuk keamanan sebagai perlindungan konsumen dalam melakukan teransaksi di industri keuangan digital sangat penting. Bentuk tanda tangan digital tidak seperti tandan tangan umumnya, hanya berbentuk keterangan. Namun, ada kode tertentu yang hanya bisa dipindai secara khusus . Jadi keamanannya sangat terjamin. Untuk mengetahui keamanannya dilakukan dengan pengujian menggunakan aplikasi *Hasher Lite*. Untuk proses keamana tanda tangan digital menggunakan metode SHA-1 (Secure Hash Algorithm-1). Diketahui tanda tangan dengan resolusi 5 x 5 pixel.

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79



Gambar 1. Tanda tangan Pixel 5 x 5

Gambar diatas adalah gambar tanda tangan digital yang diambil resolusi 5x5. Gambar tanda tangan digital diasas dikasih baground kuning dikernakan gambar diatas cuman mempunyai dua warna, putih dan hitam. Maka yang resolusi 5x5 terlihat putih tidak nampak kotak-kotak 5x5. Metode SHA-1 dikatakan aman kerena proses SHA-1 dihitung secara infisibel untuk mencari string yang sesuai menghasilkan *message digest* atau dapat juga digunakan untuk mencari dua string yang berbeda yang akan menghasilkan *message digest* yang sama. Hash merupakan suatu metode yang secara langsung mengakses record-record dalam suatu tabel dengan melalukan transformasi aritmatika pada *Key* yang menjadi alamat adalah tabel tersebut. *Key* merupakan suatu input dari pemakai dimana pada umumnya berupa nilai *string* krakter.

Tabel 1. Pixel resolusi 5 x 5

255	255	255	255	255
255	255	255	255	255
255	255	255	255	255
255	255	255	255	255
255	255	255	255	255

Pesan : 255255255255255, 255255255255255, 255255255255255, 255255255255255, 255255255255255.

Tabel 2. Input nilai pixel ke biner

11111111	11111111	11111111	11111111	11111111
11111111	11111111	11111111	11111111	11111111
11111111	11111111	11111111	11111111	11111111
11111111	11111111	11111111	11111111	11111111
11111111	11111111	11111111	11111111	11111111

Pesan yang diatas diubah menjadi biner dan proses perhitungan menggunakan metode SHA-1 dari 0 – 7 9 putaran. Langkah pertama yaitu :

1. Melakukan Penambahan pandding bit.
   200 + 1 + K = 448 mod 512
   K = 448 – 201 mod 512
   K = 247
   Lakukan penambahan bit sebanyak 247 bit.

Tabel 3. Penambahan 247 Bit

M0	11111111	11111111	11111111	11111111
M1	11111111	11111111	11111111	11111111
M2	11111111	11111111	11111111	11111111
M3	11111111	11111111	11111111	11111111
M4	11111111	11111111	11111111	11111111
M5	11111111	11111111	11111111	11111111
M6	11111111	**10000000**	**00000000**	**00000000**
M7	**00000000**	**00000000**	**00000000**	**00000000**
M8	**00000000**	**00000000**	**00000000**	**00000000**
M9	**00000000**	**00000000**	**00000000**	**00000000**
M10	**00000000**	**00000000**	**00000000**	**00000000**
M11	**00000000**	**00000000**	**00000000**	**00000000**
M12	**00000000**	**00000000**	**00000000**	**00000000**
M13	**00000000**	**00000000**	**00000000**	**00000000**

   Biner yang dihitamkan dalam tabel diatas adalah nilai dari tambahan bit sebanyak 247 bit.
2. Penambahan panjang pesan

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

Melakukan penambahan panjang pesan sebanyak 64 bit dan dibuat juga seperti tabel diatas dan tambahan nilainya dihitamkan supaya diketahui mana yang bit ditambahkan.

Tabel 4. Penambahan panjang pesan

M0	11111111	11111111	11111111	11111111
M1	11111111	11111111	11111111	11111111
M2	11111111	11111111	11111111	11111111
M3	11111111	11111111	11111111	11111111
M4	11111111	11111111	11111111	11111111
M5	11111111	11111111	11111111	11111111
M6	11111111	**10000000**	**00000000**	**00000000**
M7	**00000000**	**00000000**	**00000000**	**00000000**
M8	**00000000**	**00000000**	**00000000**	**00000000**
M9	**00000000**	**00000000**	**00000000**	**00000000**
M10	**00000000**	**00000000**	**00000000**	**00000000**
M11	**00000000**	**00000000**	**00000000**	**00000000**
M12	**00000000**	**00000000**	**00000000**	**00000000**
M13	**00000000**	**00000000**	**00000000**	**00000000**
M14	**00000000**	**00000000**	**00000000**	**00000000**
M15	**00000000**	**00000000**	**00000000**	**10001000**

3. Parsing pesan (Mengelompokan Pesan)
   Melakukan pengelompokan pesan dari penambahan bit sampai ke penambahan panjang pesan.

Tabel 5. Pengelompokan pesan

M0	11111111	11111111	11111111	11111111
M1	11111111	11111111	11111111	11111111
M2	11111111	11111111	11111111	11111111
M3	11111111	11111111	11111111	11111111
M4	11111111	11111111	11111111	11111111
M5	11111111	11111111	11111111	11111111
M6	11111111	10000000	00000000	00000000
M7	00000000	00000000	00000000	00000000
M8	00000000	00000000	00000000	00000000
M9	00000000	00000000	00000000	00000000
M10	00000000	00000000	00000000	00000000
M11	00000000	00000000	00000000	00000000
M12	00000000	00000000	00000000	00000000
M13	00000000	00000000	00000000	00000000
M14	00000000	00000000	00000000	00000000
M15	00000000	00000000	00000000	10001000

4. Penjadwalan pesan
   Wt= $\sum_{ROTl}^{Mt} (Wt-3 \oplus Wt-8 \oplus Wt-14 \oplus Wt-16)$
   Wt : Pesan Ke t yang baru
   Wt = $ROTL\ 1(W16-3 \oplus W16-8 \oplus W16-14 \oplus W16-16)$
   Wt = $ROTL\ 1(W13 \oplus W8 \oplus W2 \oplus W0)$
   Wt = ROTL 1 (00000000 ⊕ 00000000 ⊕ FFFFFFFF ⊕ FFFFFFFF)

   W13 = 00000000 00000000 00000000 00000000
   W8 = 00000000 00000000 00000000 00000000
   W2 = 11111111 11111111 11111111 11111111
   W0 = 11111111 11111111 11111111 11111111 ⊕
   W16 = 00000000 00000000 00000000 00000000
   W16 = 00000000 00000000 00000000 00000000
   W16 = 00000000 00000000 00000000 00000000
   0 0 0 0 0 0 0 0

   Dan langkah selanjutnya sampai ke putaran tujuh puluh sembilan. Dari hasil keseluruhan dari proses SHA-1 diatas , maka diperoleh hasil yang menggunakan nilai hexadesimal dari peroses hasil tersebut. Sebagaimana terlihat pada tabel dibawah ini sebagai berikut:

Tabel 6. Hasil dari biner ke Hexadesimal

W0 = FFFFFFFF	W1 = FFFFFFFF	W2 = FFFFFFFF	W3 = FFFFFFFF	W4 = FFFFFFFF
W5 = FFFFFFFF	W6 = FF800000	W7 = 00000000	W8 = 00000000	W9 = 00000000

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

W10 = 00000000	W11 = 00000000	W12 = 00000000	W13 = 00000000	W14 = 00000000
W15 = 00000088	W16 = 00000000	W17 = 00000000	W18 = 00000110	W19 = 00000000
W20 = 00FFFFFE	W21 = FFFFFDDF	W22 = FF000001	W23 = FEFFFFFD	W24 =FFFFFBBF
W25 = FE000003	W26 = FDFFFFFB	W27 = FFFFF77F	W28 =FDFFFFFB	W29 = 04000558
W30 = 01FFEFEE	W31 = 01FFEFEC	W32 = FFFFFFEF	W33=FFFFDDDB	W34 =FFFFDFDF
W35 = FFFFEA9F	W36 = FA004643	W37 = F5FFB14B	W38 =FDFFFD9F	W39 =F600ACA7
W40 = 17FF6A30	W41 = 07FFAE70	W42 = 14011900	W43 = 27FEE510	W44 = 03FFF04C
W45 = 4802111C	W47 = 17FEDDBC	W48 = BFFB9FF7	W49 = 9FFB1050	W50 = 0C00BA40
W51 = 240A4267	W51 = 37F69781	W52 = 37F69781	W53 = 8FFA44AF	W54 = 341181A9
W55 = A3EE91D5	W56 = 67FF50D0	W57 = 17D7B533	W58 = 782185B3	W59 = 5812E4B9
W60 = D7B9C7C4	W61 = 504F32B6	W62 = C7F7DFF7	W63 = F8A53E7B	W64 = 08968FA3
W65 = 37A261E7	W66 = 76E53E13	W67 =F6E8DBA5	W68 = C7F96017	W69 =157DB3BE
W70 = 25E3BAD7	W71 = 16CAE715	W72 = 046BD2FE	W73 = BB091574	W74 = 9F6F36E3
W75 = F5BDBEA	W76 = 397CCAA1	W77 = 45F10120	W78 = DE7EB855	W79 = C162F651

5. Menginisialisasi 5 variabel kerja a, b, c, d dan e

Tabel 7. Menginisialisasi

a = 67452301	b = efcddab89	c = 98badcfe	d = 10325476	e = c3d2e1f0

$T = ROTL^5(a) + f_t(b.c,d) + e + Kt + Wt$
$e = d$
$d = c$
$c = ROTL^{30}(b)$
$b = a$
$a = T^1$

$f_t(b,c,d)$:

Ch	(b,cd) = (b Λ c)⊕ (b Λ d)	0 ≤	t ≤ 19
Parity	(b,c,d) = b ⊕ c ⊕ d	20 ≤	t ≤ 39
Maj	(b,c,d) = (b Λ c)⊕ (b Λ d) ⊕ (c Λ d)	40 ≤	t ≤ 59
Parity	(b,c,d) = b ⊕ c ⊕ d	60 ≤	t ≤ 79

Kt:

= 5A027999	0 ≤t ≤ 19
= 6ED9EBA1	20 ≤t ≤ 39
= 8F1BBCDC	40 ≤t ≤ 59
= CA62C1D6	60 ≤t ≤ 79

$T = ROTL^5(a) + f_t(b.c,d) + e + Kt + Wt$
Dan selanjutnya mencari nilai dari Ch, Parity, Maj dan Parity

Ch (b, c, d) = ((b Λ c) ⊕ (b Λ d)
= EFCDAD89 Λ 98BADCFE ⊕
= EFCDAD89 Λ 10325476
= 88888C88 ⊕ 00000000
= 88888C88

Parity (b,c, d) = (b ⊕ c ⊕ d)
= EFCDAB89 ⊕ 98BADCFE ⊕ 10325476
= 77777777 ⊕ 10325476
= 67452301

Maj (b, c, d) = (b Λ c)⊕ (b Λ d) ⊕ (c Λ d)
= EFCDAB89 ⊕ 98BADCFE
= EFCDAB89 ⊕ 10325476
= 98BADCFE ⊕ 10325476
= 88888888 ⊕ 00000000 ⊕ 10325476
= 98BADCFE

Parity (b, c, d) = b ⊕ c ⊕ d
= EFCDAB89 ⊕ 98BADCFE ⊕ 10325476
= 77777777 ⊕ 10325476
= 67452301

Nilai (a) akan ROTL sebanyak 5 x kekiri dan jumlahkan dengan nilai Ch, Parity ,Maj, Parity, Kt dan Wt untuk mencari nilai A dan (b) akan di ROTL kan sebanyak 30 x kekiri untuk mencari nilai C. masuk ke penyelesaian menginilisasi 5 variabel tersebut untuk mencari nilai dari A dan C.
Langkah pertama menggunakan nilai Ch = 88888c88 dan Kt = 5a827999 dari 0 ≤ 19

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

T = ROTL^5 (a) + ft (b. c, d) + e + Kt+ Wt
ROTL^5 (a) = 67452301
= E8A46020 + 88888C88 + 5A027999 +FFFFFFFF
= CB2F6640
e = d = 10325476
d = c = 98BADCFE
ROTL^{30} (b) = EFCDAB89
= 11101111 10111101 10101011 10001001
= 01111011 11101111 01101010 11100010
7 B E F 6 A E 2
b = a = 6452301
a = t0 = CB2F6640

Dan langkah selanjutnya sampai ke putaran sembilan belas.
Dan langkah selanjutnya menggunakan nilai Prity = 67452301 dan Kt =6ed9eba1 dari 20≤ 39
ROTL^5 (a) = CFA4F041
= F49E08020+ 67452301+ 6ED9EBA1 + 00FFFFFE
= CBBD16C0
e = d = 10325476
d = c = 98BADCFE
ROTL^{30} (b) = 89EFBDAB
= 10001001 11101111 10111101 10101011
= 11100010 01111011 11101111 01101010
E 2 7 B E F 6 A
b = a = CFA4F041
a = t20 = CBBD16C0

Dan langkah selanjutnya sampai ke putaran lima puluh sembilan.
Dan langkah selanjutnya menggunakan nilai parity = 67452301 dan nilai Kt = ca62c1d6 dari 60 ≤ 79
ROTL^5 (a) = 2E6EA433
= CDD48660 + 67452301 + CA62C1D6 + D7B9C7C4
= D73632FB
e = d = 10325476
d = c = 98BADCFE
ROTL^{30} (b) = BDAB89EF
= 10111101 10101011 10001001 11101111
= 11101111 01101010 11100010 01111011
E F 6 A E 2 7 B
b = a = 2E6EA433
a = t60 = D73632FB

Dan langkah selanjutnya samapai keputaran tujuh puluh sembilan. Setelah melakukan perhitungan menginilisasi 5 variabel tersebut dengan sebanyak 79 putaran tersebut, guna untuk mencari nilai dari A dan C. Dari hasil keseluruhan dari proses SHA-1 diatas, maka diperoleh hasil yang menggunakan nilai Hexadesimal dari proses hasil tersebut. Sebagai bagaimana terlihat pada tabel dibawah ini sebagai berikut:

Tabel 8. hasil Menginisialisasi 5 variabel

	a	b	c	d	e
T	B6E15969	EFCDAB89	98BADCFE	10325476	C3d2e1f0
T0	CB2F6640	67452301	7BEF6AE2	98BADCFE	10325476
T1	4877CE20	EFCDAB89	9EFBDAB8	10325476	98BADCFE
T2	F184CA20	67452301	27BEF6AE	98BADCFE	10325476
T3	13244A20	EFCDAB89	89EFBDAB	10325476	98BADCFE
T4	47144A20	67452301	E27BEF6A	98BADCFE	10325476
T5	C5144A20	EFCDAB89	B89EFBDA	10325476	98BADCFE
T6	84944A21	67452301	AE27BEF6	98BADCFE	10325476
T7	75144A41	EFCDAB89	AB89EFBD	10325476	98BADCFE
T8	85144E41	67452301	6AE27BEF	98BADCFE	10325476
T9	8514CE41	EFCDAB89	DAB89EFB	10325476	98BADCFE
T10	8524CE41	67452301	F6AE27BE	98BADCFE	10325476
T11	8724CE41	EFCDAB89	BDAB89EF	10325476	98BADCFE
T12	C724CE41	67452301	EF6AE27B	98BADCFE	10325476
T13	C724CE41	EFCDAB89	FBDAB89E	10325476	98BADCFE
T14	C724CE41	67452301	BEF6AE27	98BADCFE	10325476
T15	C724CEC9	EFCDAB89	EFBDAB89	10325476	98BADCFE

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

T16	C724DF41	67452301	7BEF6AE2	98BADCFE	10325476
T17	C726EE41	EFCDAB89	9EFBDAB8	10325476	98BADCFE
T18	C768CF51	67452301	27BEF6AE	98BADCFE	10325476
T19	CFA4F041	EFCDAB89	89EFBDAB	10325476	98BADCFE
T20	CBBD16C0	67452301	E27BEF6A	98BADCFE	10325476
T21	4DC1E481	EFCDAB89	B89EFBDA	10325476	98BADCFE
T22	8D5B9EC3	67452301	AE27BEF6	98BADCFE	10325476
T23	8092E6FF	EFCDAB89	AB89EFBD	10325476	98BADCFE
T24	E87BEA41	67452301	6AE27BEF	98BADCFE	10325476
T25	E39C56C5	EFCDAB89	DAB89EFB	10325476	98BADCFE
T26	47A9E73D	67452301	F6AE27BE	98BADCFE	10325476
T27	CB5BEDC1	EFCDAB89	BDAB89EF	10325476	98BADCFE
T28	3F9CC6BD	67452301	EF6AE27B	98BADCFE	10325476
T29	CDB7EB9A	EFCDAB89	FBDAB89E	10325476	98BADCFE
T30	8F1C71D0	67452301	BEF6AE27	98BADCFE	10325476
T31	BBAD388E	EFCDAB89	EFBDAB89	10325476	98BADCFE
T32	4BC62051	67452301	7BEF9AE2	98BADCFE	10325476
T33	4EE2F69D	EFCDAB89	9EFBDAB8	10325476	98BADCFE
T34	B27DC221	67452301	27BEF6AE	98BADCFE	10325476
T35	25D73D61	EFCDAB89	89EFBDAB	10325476	98BADCFE
T36	8B070105	67452301	E27BEF6A	98BADCFE	10325476
T37	D2FEE08D	EFCDAB89	B89EFBDA	10325476	98BADCFE
T38	33FB1DE1	67452301	AE27BEF6	98BADCFE	10325476
T39	4B837769	EFCDAB89	AB89EFBD	10325476	998BADCFE
T40	C454F12A	67452301	6AE27BEF	98BADCFE	10325476
T41	BA7A6D8A	EFCDAB89	DAB89EFB	10325476	98BADCFE
T42	8A65641A	67452301	F6AE27BE	98BADCFE	10325476
T43	9C82022A	EFCDAB89	BDAB89EF	10325476	98BADCFE
T44	BC16CF66	67452301	EF6AE27B	98BADCFE	10325476
T45	72B30C95	EFCDAB89	FBDAB89E	10325476	98BADCFE
T46	C63A3D96	67452301	BEF6AE27	98BADCFE	10325476
T47	071D2A56	EFCDAB89	EFBDAB89	10325476	98BADCFE
T48	CB778491	67452301	7BEF6AE2	98BADCFE	10325476
T49	36C23C4A	EFCDAB89	9EFBDAB8	10325476	98BADCFE
T50	0C1EDD5A	67452301	27BEF6AE	98BADCFE	10325476
T51	CFBC8781	EFCDAB89	89EFBDAB	10325476	98BADCFE
T52	575E217B	67452301	E27BEF6A	98BADCFE	10325476
T53	A3950DE9	EFCDAB89	B89EFBDA	10325476	98BADCFE
T54	BF89D8A3	67452301	AE27BEF6	98BADCFE	10325476
T55	BD00400F	EFCDAB89	AB89EFBD	10325476	98BADCFE
T56	2FDDEC8A	67452301	6AE27BEF	98BADCFE	10325476
T57	3B6BE04D	EFCDAB89	DAB89EFB	10325476	98BADCFE
T58	0D74292D	67452301	F6AE27BE	98BADCFE	10325476
T59	2E6EA433	EFCDAB89	BDAB89EF	10325476	98BADCFE
T60	D73632FB	67452301	EF6AE27B	98BADCFE	10325476
T61	68BD76ED	EFCDAB89	FBDAB89E	10325476	98BADCFE
T62	114EA26E	67452301	BEF6AE27	98BADCFE	10325476
T63	54217112	EFCDAB89	EFBDAB89	10325476	98BADCFE
T64	BE6C96BA	67452301	7BEF6AE2	98BADCFE	10325476
T65	36DD1DFE	EFCDAB89	9EFBDAB8	10325476	98BADCFE
T66	8430E2AA	67452301	27BEF6AE	98BADCFE	10325476
T67	AEAD15BC	EFCDAB89	89EFBDAB	10325476	98BADCFE
T68	CF43FC6E	67452301	E27BEF6A	98BADCFE	10325476
T69	2FA52655	EFCDAB89	B89EFBDA	10325476	98BADCFE
T70	4C306A4E	67452301	AE27BEF6	98BADCFE	10325476
T71	CE8015AC	EFCDAB89	AB89EFBD	10325476	98BADCFE
T72	06166D55	67452301	6AE27BEF	98BADCFE	10325476
T73	AF7EA4EB	EFCDAB89	DAB89EFB	10325476	98BADCFE
T74	C0EBB91A	67452301	F6AE27BE	98BADCFE	10325476
T75	44DCC6C0	EFCDAB89	BDAB89EF	10325476	98BADCFE
T76	6BD8778	67452301	EF6AE27B	98BADCFE	10325476
T77	4F49D4F7	EFCDAB89	FBDAB89E	10325476	98BADCFE

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 3, Juni 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i3.2833
Hal 71−79

T78	F9613C0	67452301	BEF6AE27	98BADCFE	10325476
T79	1F325CA8	EFCDAB89	EFBDAB89	10325476	98BADCFE

Setelah menginisialisasi 5 variabel tersebut, selanjtnya hasil akhir dari perhitungan dijumlahkan dengan nilai pertama :

$H_0^{(1)}$ = 1F325CA8 + 67452301 = CFBC8980
$H_0^{(2)}$ = EFCDAB89 + EFCDAB89 = 9b81A951
$H_0^{(3)}$ = EFBDAB89 + 98BADCFE = 465BF0DD
$H_0^{(4)}$ = 10325476 + 10325476 = 5EAC7570
$H_0^{(5)}$ = 98BADCFE + C3D2E1F0 = 8AAE267D

Tabel 9. Hasil nilai SHA-1

CFBC8980
9b81A951
465BF0DD
5EAC7570
8AAE267D

Berdasrkan dari perhitungan diatas diperoleh nilai SHA-1 berbentuk bilangan Hexadesimal 5 karakter 40 byte, yaitu :" CFBC8980, 9b81A951, 465BF0DD, 5EAC7570, 8AAE267D".

3.1 Hasil Pengujian

Aplikasi *Hasher Lite* aplikasi pengamanan data. Fungsi *hash* untuk menjaga integritas bukti digital. Jadi ketika ada sebuah kasus yang melibatkan perangkat digital, perangkat digital tersebut dilakukan penyitaan, maka harus dijaga keaslian barang bukti tersebut. Pada perangka digital, salah satu langkah untuk menjaga integritas dan keaslian bukti digital tersebut yaitu dengan fungsi *hash*. Metode SHA-1 untuk mengamanakan tanda tangan digital maka didapat sebuah hasil sebagai berikut:

Tabel 10. Tabel Hasil Pengujian

Parameter	Tanda Tangan Asli	Tanda Tangan Manipulasi	Nilai Hash Asli	Nilai Hash Manipulasi
Melakukan perubahan pada tanda tagan	 Item type:JPG Dimensions:616x857 Size:85,9 KB	 Item type:JPG Dimensions:196x196 Size:16,2 KB	86777FA90F9B571288788 8872964A8EC5C8DBEEE	a4c913673ccc8ce2efc5c 7c572330b2d 8f3d649e
Melakukan penambahan *Frame* pada Tanda tangan	 Item type:JPG Dimensions:616x857 Size:85,9 KB	 Item type:JPG Dimensions:196x196 Size:16,2 KB	86777FA90F9B571288788 8872964A8EC5C8DBEEE	a4c913673ccc8ce2efc5c 7c572330b2d 8f3d649e
Merubah ekstensi tanda tangan	 Item type:JPG Dimensions:616x857 Size:85,9 KB	 Item type:JPG Dimensions:196x196 Size:16,2 KB	86777FA90F9B571288788 8872964A8EC5C8DBEEE	a4c913673ccc8ce2efc5c 7c572330b2d 8f3d649e

4. KESIMPULAN

Dari hasil penulisan dan analisa yang dilakukan dapat disimpulkan bahwa keamanan pada tanda tangan digital bertujuan untuk mengamankan tanda tangan dari suatu dokumen dari orang-orang yang tidak bertangung jawab atau tidak berhak dalam dokumen tersebut dengan menggunakan metode SHA-1. Metode SHA-1 suatu peroses perhitungan nilai-nilai pixel dari tanda tangan untuk mengamakan pada tanda tangan digital dengan resolusi 5 x 5 pixel.

REFERENCES

[1] J. Arifin and M. Z. Naf'an, "Verifikasi Tanda Tangan Asli Atau Palsu Berdasarkan Sifat Keacakan (Entropi)," J. Infotel, vol. 9, no. 1, p. 130, 2017.
[2] K. Aryasa and Y. T. Paulus, "Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java," vol. 1, no. 1, pp. 57–66, 2014.

[3] D. Prof.Dr.jogiyanto HM, MBA., Ph, "Analisis & Desain Sistem Informasi," in Analisis & Desain Sistem Informasi, Yogyakarta: C.V ANDI OFFSET, 2005, p. 129.
[4] R. Sadikin, Kriptografi keamanan jaringan dan implementasinya dalam bahasa java. Yogyakarta: CV.ANDI OFFSET, 2012.
[5] M. K. Emy Setyaningsih, S.Si., Kriptografi & Implementasinya Menggunakan Matlab. Yogyakarta: CV.ANDI OFFSET, 2015.
[6] H. Hasrul and L. H. Siregar, "Penerapan Teknik Kriptografi pada Database menggunakan Algoritma One Time Pad," Elektron. Sist. Inf. dan Komput., vol. 2, no. 2, pp. 41–52, 2016.
[7] M. Ir. Yusuf Kurniawan, Kriptografi keamanan internet dan jaringan komunikasi. Bandung: Informatika Bandung, 2004.