Jurnal CyberTech

Vol.x. No.x, Maret 2019, pp. xx~xx
P-ISSN : 9800-3456
E-ISSN : 2675-9802

Implementasi *Digital Signature* Untuk Validasi Dokumen *E-Invoice* Teknoweb Indonesia Dengan Penerapan Algoritma SHA (*Secure Hash Algorithm*) Dan DSA (*Digital Signature Algorithm*)

Radika Maidinda *, Dicky Nofriansyah **, Masyuni Hutasuhut **

* Sistem Informasi, STMIK Triguna Dharma
** Sistem Informasi, STMIK Triguna Dharma

Article Info

Article history:
Received Jun 12th, 2021
Revised Aug 20th, 2021
Accepted Aug 26th, 2021

Keyword:
Digital signature
Tanda tangan digital
SHA *(Secure Hash Algorithm)*
DSA *(Digital Signature Algorithm)*
Fungsi *Hash*

ABSTRACT

Dokumen *e-invoice* merupakan tagihan pemesan dalam jual beli baik jasa maupun produk yang menjadi sebuah kebutuhan, hal ini akan mempermudah pihak Teknoweb Indonesia untuk mendistribusikan dokumen melalui berbagai media komunikasi elektronik. Dengan adanya kemudahan tersebut, maka *authentication* dan *data integrity* merupakan hal yang sangat penting untuk menjaga kerahasiaan dan keamanan data dokumen *e-invoice* dari pihak tidak bertanggung jawab yang turut berkomunikasi guna memanfaatkan data untuk kepentingan pribadi. Pada permasalahan tersebut adapun cara untuk melakukan tindakan pencegahan yaitu dengan mengubah pesan menjadi sebuah kode. Ilmu pengetahuan yang dapat diterapkan untuk menjaga *authentication* dan *data integrity* tetap dalam keadaan aman yaitu kriptografi pada *digital signature*. Hasil penelitian merupakan terbentuknya sistem aplikasi digital signature untuk memvalidasi keabsahan dokumen *e-invoice* dengan penerapan metode SHA (*Secure Hash Algorithm*) dan DSA (*Digital Signature Algorithm*). Pemanfaatan algoritma SHA-1 akan melindungi pesan saat proses distribusi yaitu dengan menghitung nilai *hash* pada dokumen *e-invoice* dan algoritma DSA dapat memberikan jaminan *othentikasi* pengirim maupun penerima dokumen.

Corresponding Author: *First Author
Radika Maidinda
Sistem Informasi
STMIK Triguna Dharma
Email: radikamaidinda11@gmail.com

1. PENDAHULUAN

Digital Signature atau tanda tangan digital merupakan salah satu perkembangan utama dalam dunia keamanan jaringan dan data. Kebutuhan akan *digital signature* terus mengalami peningkatan seiring dengan pertumbuhan komunikasi digital. Algoritma tanda tangan digital mengautentikasi integritas dari data yang

ditandatangani dan identitas dari penandatangan. Autentikasi tanda tangan digital merupakan proses yang dilakukan dimana penerima pesan digital dapat mempercayai integritas dari pesan dan pengirim.

Faktanya tanda tangan digital telah diimplementasikan di berbagai sektor bisnis, terutama terkait hal pemerintahan. Pertama kalinya pemerintahan di Amerika Serikat menerbitkan versi elektronik baik dari hukum secara umum maupun privat. Universitas Chicago, juga telah menerapkan transkrip nilai mahasiswa yang disertai tanda tangan[1]. Hal ini membuktikan bahwa teknik autentikasi sangat diperlukan untuk pesan dalam bentuk manual maupun digital.

Disebabkan dengan kemajuan teknologi komunikasi yang ada, terdapat pula pihak-pihak yang tidak dikehendaki dengan sengaja ikut berkomunikasi, dengan kata lain terdapat pihak yang tidak bertanggung jawab turut memanfaatkan pesan guna kepentingan pribadi. Sehingga pihak tersebut dapat mengetahui isi pesan maupun dapat mengubah pesan. Beberapa masalah tesebut dialami oleh salah satu perusahaan Teknoweb Indonesia yang bergerak di bidang ICT pada pertengahan tahun 2020 yaitu tingkat keamanan *e-invoice* yang dikeluarkan sebagai bukti transaksi masih terbilang sangat rendah, yaitu dengan kurangnya hal yang membuktikan keaslian dari *e-invoice* dari pihak administrasi. Oleh karena itu, masalah keamanan data merupakan suatu aspek yang penting dari suatu data terutama untuk keaslian dan integritas dari data atau pesan tersebut.

Oleh karena itu diperlukan suatu cara untuk menjaga pesan yang terdapat pada *e-invoice* tetap dalam keadaan asli hingga sampai kepada penerima. Adapun cara untuk melakukan tindakan pencegahan yaitu dengan mengubah pesan menjadi sebuah kode yang hanya dapat dipahami oleh pengirim dan penerima pesan. Ilmu pengetahuan yang dapat diterapkan untuk menjaga kerahasiaan pesan tetap dalam keadaan aman adalah kriptografi. Kriptografi adalah untuk menjaga rahasia plaintekt (kunci atau kedua-dua) dari seorang penyusup, yang disebut musuh (*adversaries*), penyerang (*attacker*), penyusup (*interceptors*), penyelundup (*interlopers*), pengacau(*intrud*ers), lawan (*oppnents*) yang diasumsikan memiliki akses penuh untuk berkomunikasi antara pengirim dan penerima[2].

Kriptografi merupakan teknik dalam bidang ilmu matematika yang berhubungan dengan hal-hal terkait informasi seperti integritas data, kerahasiaan serta otentikasi guna aspek keamanan[3]. Namun, tidak cukup dengan mengubah pesan menjadi sandi karena tidak menutup kemungkinan pesan tetap dapat diubah oleh pihak ke tiga. Untuk memperkuat kerahasiaan serta keabsahan dari pesan tersebut, yaitu menggunakan sebuah *digital signature* atau disebut dengan tanda tangan digital yang memodifikasi dari sistem kriptografi kunci publik (*public key*), sama hal seperti tanda tangan manual hanya saja perbedaannya yaitu pengirim menyertakan tanda tangan berupa kode sandi dalam bentuk *string* yang terbentuk dari kunci publik dan kunci privat yang telah ditentukan berdasarkan dengan pesan yang akan dikirim. Maka tanda tangan itulah yang nanti dapat digunakan untuk memverifikasi keabsahan pesan pada *e-invoice* Teknoweb Indonesia.

Ada beberapa algoritma dalam pembentukan tanda tangan digital, diantaranya: RSA (*Rivest-Shamir-Adleman*) *Signature Scheme*, ElGamal *Signature Scheme*, *Schnorr Signature Scheme*, dan DSA (*Digital Signature Algorithm*). Adapun terdapat perbedaan yang sangat mendasar diantara skema tersebut salah satu yaitu pada proses pembentukan tanda tangan DSA (*Digital Signature Algorithm*) yang membutuhkan penambahan fungsi *hash* pada proses penandatanganan yang akan digunakan untuk mereduksi pesan asli menjadi suatu *message digest* (nilai *hash*) yang berupa *string* pendek dengan panjang tetap sesuai ketentuan masing-masing[4].

Kedepanya hal baru yang ada di Teknoweb Indonesia adalah sistem *digital signature* berbasis web yang mengadopsi Metode SHA (*Secure Hash Algoritm*) dan DSA (*Digital Signature Algorithm*) yang mampu menyelesaikan masalah khusus validasi keabsahan dokumen *e-invoice*. Berdasarkan kondisi tersebut maka penulis mengangkat judul penelitian yaitu: "**Implementasi *Digital Signature* Untuk Validasi Dokumen *E-Invoice* Teknoweb Indonesia Dengan Penerapan Algoritma Sha (*Secure Hash Algorithm*) Dan Dsa (*Digital Signature Algorithm*)**".

2. METODE PENELITIAN

Metode yang digunakan adalah model spiral. Model spiral merupakan salah satu diantara bentuk evolusi dengan metode iterasi natural yang mengadopsi dua model perangkat lunak yaitu penggabungan model *prototyping* dengan aspek sistimatis yang merupakan pengembangan dari model *waterfall*[5].

Terdapat 6 tahap wilayah tugas dalam model pengembangan sistem spiral, antara lain :

1. *Liaison* (Komunikasi Pelanggan)
   Identifikasi atau mengkomunikasikan antara pihak administrasi keuangan dan kebutuhan-kebutuhan yang terdapat pada *e-invoice*, serta yang di butuhkan di dalam sistem *digital signature*.
2. *Planning* (Perencanaan)
   Kegiatan untuk menetapkan tujuan dari sistem yang akan dibangun beserta cara-cara untuk mencapai tujuan dari sistem tersebut.
3. *Risk Analysis* (Analisis Resiko)
   Merupakan keadaan ketidakpastian mengenai suatu keadaan yang akan terjadi di masa depan berdasarkan keputusan atau pun tindakan yang diambil dengan berbagai pertimbangan pada saat ini.
4. *Engineering* (Perekayasaan)
   Hal-hal yang dibutuhkan untuk melakukan pembangunan guna mewakili satu atau lebih dari sistem tersebut.
5. *Construction and release* (Konstruksi dan Peluncuran)
   Merupakan tugas yang dibutuhkan untuk melakukan perancangan, pengujian, pemasangan dan pemberian pelayanan kepada pengguna yaitu administrasi keuangan Teknoweb Indonesia.
6. *System Evaluation* (Evaluasi Sistem)
   Yaitu tugas-tugas untuk mendapatkan umpan balik dari pengguna sebagai bahan evaluasi.

3. ANALISA DAN HASIL

Adapun dalam analisa ini menggunakan kombinasi dari dua metode diantaranya yaitu SHA-1 dan DSA. Pemanfaatan algoritma SHA-1 akan melindungi pesan saat proses distribusi yaitu dengan menghitung nilai *hash* pada dokumen *e-invoice* dan algoritma DSA dapat memberikan jaminan *othentikasi* pengirim maupun penerima dokumen.

3.1. SHA-1 (*Secure Hash Algorithm)*

SHA (*Secure Hash Algorithm)* merupakan fungsi *hash* satu-arah yang diciptakan oleh NIST (*National Institute of Standards and Technology*) digunakan bersama DSS (*Digital Signature Standard*). Oleh NSA (*National Security Agency*) telah menyatakan bahwa SHA digunakan sebagai standard fungsi *hash* satu-arah yang didasarkan pada MD5 yang dibuat oleh Ronald L. Rivest dari MIT (*Massachusetts Institute of Technology*). Adapun SHA terbagi menjadi 5 yaitu SHA-1,SHA-224, SHA-256, SHA-384, SHA-512[2],[6].

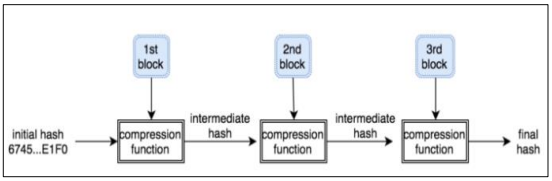
Tabel 1**.** Macam-macam SHA

Algoritma		**Naskah (bit)**	**Blok (bit)**	**Kata (bit)**	**Nilai *hash*/ *message digest* (bit)**
SHA-1	SHA-1	<264	512	32	160
SHA-2	SHA-256 / 224	<264	512	32	256
	SHA-384	<2128	1024	64	384
	SHA-521	<2128	1024	64	512
SHA-3	SHA-3 (*Keccak*)	Beragam	Beragam	Beragam	Beragam

penulis menggunakan fungsi SHA-1 dikarenakan memiliki nilai *hash* terkecil diantara lainnya dan lebih panjang dari MD5. SHA-1 dapat menerima masukan berupa pesan dengan ukuran panjang maksimum 264 bit atau setara dengan 2.147.483.648 gigabyte dan dapat menghasilkan berupa nilai *hash* dengan panjang 160 bit. Nilai *hash* tersebut yang nantinya akan digunakan kedalam DSA untuk menghitung tanda tangan digital

pada sebuah pesan. Nilai *hash* pada pesan yang diperoleh oleh *receiver* akan menghasilkan nilai yang sama dengan *sender,* saat menghitung pesan tersebut pada fungsi SHA-1[2],[4],[7].

Pada SHA-1 tidak dapat menemukan dua pesan yang berbeda menghasilkan nilai *hash* yang sama atau tidak mungkin menemukan pesan aslinya jika diberikan suatu nilai *hash*-nya. Pesan (M) dengan panjang (L) bit yaitu $1 \leq L \leq 2^{64}$. Seperti halnya MD5, algoritma SHA-1 pun sudah ditemukan kolisinya. yaitu Rijmen dan dan Oswald yang pertama kali mempublikasikan serangan pada versi SHA-1 yang direduksi (hanya menggunakan 53 putaran dari 80 putaran) pada tahu 2005 dan menemukan kolisi dengan kompleksitas sekitar 280 operasi. Pada bulan Februari 2005, Xiayoun Wang, Yiqun Lisa Yin, dan Hongbo Yo mempublikasikan serangan yang dapat menemukan kolisi pada versi penuh SHA-1 dan membutuhkan sekitar 264 operasi [8]. Rizki Wicaksono, seorang *hacker* alumni informatika ITB mendemonstrasikan cara membentuk dua file PDF berbeda dengan nilai *hash* SHA-1 yang dihasilkan sama[9].



Gambar 1. Contoh *Hash* SHA-1 Memproses 3 Blok *Input*

3.2. DSA (*Digital Signature Algorithm)*

Pada bulan Agustus 1991, NIST mengumumkan bakuan (*Standard*) untuk sebuah tanda tangan digital yang disebut sebagai DSS (*Digital Signature Standard*). DSS merupakan standart sedangkan DSA (*Digital Signature Agorithm*) merupakan algoritma. Dimana suatu standart tersebut menggunakan algoritma DSA untuk penandatanganan pesan dan SHA digunakan untuk membangkitkan *message digest* yang diperoleh dari pesan[10].

DSS terbagi menjadi dua komponen, yaitu :

1. Algorima tanda tangan digital yang disebut dengan DSA (*Digital Signature Algorithm*).
2. Fungsi *hash* standart yaitu SHA (*Secure Hash Algorithma*).

Adapun batasan bahwa nilai p pada pesan mempunyai panjang 512 sampai dengan 1024 bit dan q mempunyai panjang 160 bit. Hal ini menyebabkan DSA hampir tidak mungkin diimplementasikan dalam perangkat lunak. Panjang bit yang besar ini dimaksudkan agar upaya untuk memecahkan parameter yang lain sangat sulit dilakukan. *Compiler c* hanya sanggup menyatakan bilangan bulat hingga 2^{32}. Oleh karena itu, bila DSA diimplementasikan dalam perangkat lunak, batasan panjang bit p dan q diubah hingga maksimal nilai p dan q adalah 2^{32}[3]. DSA menggunakan fungsi *hash* SHA yang bersifat satu-arah untuk mengubah pesan asli menjadi pesan yang berukuran 160 bit[11].

3.3 Penerapan Dengan Metode

Berikut ini adalah data yang digunakan sebagai sampel dalam penelitian yaitu[12]:

Tabel 2. Data Awal

No. Invoice	*Amount*
206	1.900.000

Data awal yang diperoleh selanjutnya akan dilakukan tahap *hashing* dengan penerapan SHA-1, yaitu :

1. *Padding* (Penambahan Pesan)

   K + [panjang pesan] + 1 ≡ 448 mod 512……..[3.1]

Pada data sampel *No.Invoice* (M) memiliki sebuah pesan teks “206” dengan nilai ASCII “50 48 54” memiliki panjang 8x8=64 bit, diikuti penambahan bit “1” dan bit “0” sebanyak 448 – (64+1) = **383** (Kongruen) [4],[13]**,** kemudian Tambahkan 64 bit representasi dari panjang pesan asli dalam bentuk biner.

Tabel 3. Hasil *Padding* Pesan

Padding Pesan							
00110101	00110000	00100000	00110100	00111000	00100000	00110101	00110100
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	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	01000000

2. *Parsing* (Penguraian pesan)

Selanjutnya uraikan pesan(M) kedalam N blok yang terdiri dari 512 bit, $Y^{(0)}, Y^{(1)} \dots . Y^{(N)}$. Setiap blok *input* dari 512 bit terdiri dari 32 bit kata. Masing-masing kata yang terdiri dari 32 bit mencakup 16 bit kata. 32 bit pertama dari blok pesan *i* dinotasikan ${W_0}^{(i)}$, 32-bit berikutnya ${W_1}^{(i)}$ dan seterusnya sampai ${W_{15}}^{(i)}$ [2],[3]. Namun dalam kasus ini panjang pesan yang dihasilkan tidak lebih dari 512 bit, sehingga hanya menghasilkan 1 blok yaitu $Y^{(0)}$. Adapun tahap selanjutnya adalah membagi blok menjadi 32 bit mencakup 16 bit kata sebagai berikut :

Tabel 4. Hasil *Parsing* Pesan

Parsing Pesan			
W_0	00110101001100000010000000110100	W_8	00000000000000000000000000000000
W_1	00111000001000000011010100110100	W_9	00000000000000000000000000000000
W_2	10000000000000000000000000000000	W_{10}	00000000000000000000000000000000
W_3	00000000000000000000000000000000	W_{11}	00000000000000000000000000000000
W_4	00000000000000000000000000000000	W_{12}	00000000000000000000000000000000
W_5	00000000000000000000000000000000	W_{13}	00000000000000000000000000000000
W_6	00000000000000000000000000000000	W_{14}	00000000000000000000000000000000
W_7	00000000000000000000000000000000	W_{15}	00000000000000000000000001000000

Selanjutnya W_{16} sampai dengan W_{79} dihasilkan dari persamaan berikut :

$W_t = ROTL^1(W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3})$ ....[3.2]

Tabel 5. *Message Schadule* (W_t)

W_t (Hexadesimal)							
W_{16}	6a604069	W_{32}	754ccc73	W_{48}	73cc5151	W_{64}	1105acac
W_{17}	70406a68	W_{33}	8383cf40	W_{49}	e34d9809	W_{65}	ecdc52bc
W_{18}	00000081	W_{34}	101a901a	W_{50}	5f834c94	W_{66}	dd1e7c9c
W_{19}	d4c080d2	W_{35}	43189357	W_{51}	ce6076c1	W_{67}	be84e249
W_{20}	e080d4d0	W_{36}	9d097d65	W_{52}	46112059	W_{68}	1c76bf26
….	…….	….	……..	…..	………	…..	……..
W_{31}	6a8ecf1d	W_{47}	defaf3a5	W_{63}	582aa5a8	W_{79}	ee33e6ef

3. Penetapan Nilai Awal

Pada SHA-1 terdapat 5 nilai *buffer* juga dapat disebut dengan nilai awal atau nilai penyangga yang akan diproses dengan pesan. Penetapan nilai hash awal H_0 yaitu 32-bit kata sebanyak lima buah, dalam heksadesimal yang terdiri dari 8 karakter sebagai berikut :

$a = H_0$ = 67452301 (01100111010001010010001100000001)

$b = H_1$ = EFCDAB89 (11101111110011011010101110001001)

$c = H_2$ = 98BADCFE (10011000101110101101110011111110)
$d = H_3$ = 10325476 (00010000001100100101010001110110)
$e = H_4$ = C3D2E1F0 (11000011110100101110000111110000)

4. Pengolahan Pesan Dalam Blok Berukuran 512 bit

Adapun operasi dasar pada setiap blok yaitu :

- e ← d
- d ← c
- c ← CLS_{30}(b)
- $b \leftarrow a$
- $a \leftarrow (CLS_5(a) + f_t(b,c,d) + e + W_t + K_t)$,

Keterangan :

- a, b, c, d, e = Lima buah *buffer*
- t = Putaran, $0 \leq t \leq 79$
- f_t = Fungsi logika dengan operasi *bitwise*

Tabel 6. Fungsi Logika f_t pada setiap putaran

Putaran	$f_t(b,c,d)$	K_t
0 .. 19	$(b \wedge c) \vee (\sim b \wedge d)$	5a827999
20 .. 39	$b \oplus c \oplus d$	6ed9eba1
40 .. 59	$(b \wedge c) \vee (b \wedge d) \vee (c \wedge d)$	8f1bbcdc
60 .. 79	$b \oplus c \oplus d$	ca62c1d6

- CLS_s = *Circular Left Shift* sebanyak s bit
- W_t = *word* diturunkan dari blok dengan panjang 512 bit dari 32 bit yang sedang diproses.
- K_t = Konstanta penambah.
- $+$ = Operasi penjumlahan modulo 2^{32}

Tabel 7. Hasil Setiap Putaran

	A	B	C	D	E
Init	67452301	efcdab89	98badcfe	10325476	c3d2e1f0
T=0	d4e4b8e7	67452301	7bf36ae2	98badcfe	10325476
T=1	3b681f3b	d4e4b8e7	59d148c0	7bf36ae2	98badcfe
…	…	…	….	…	…
T=79	e3b56c9d	a0a7c386	f8df354c	0dc965a3	fa2d6cec

Selanjutnya lakukan penjumlahan hasil putaran dengan *buffer* yaitu sebagai berikut :

H_0 = 67452301 + e3b56c9d = 4afa8f9e
H_1 = efcdab89 + a0a7c386 = 90756f0f
H_2 = 98badcfe + f8df354c = 919a124a
H_3 = 10325476 + 0dc965a3 = 1dfbba19
H_4 = c3d2e1f0 + fa2d6cec = be004edc

Sehingga akan menghasilkan 160 bit *message digest* dengan lebar 20 *byte* dan menghasilkan 40 digit karakter. Adapun *message digest* dari "206" yaitu **4afa8f9e 90756f0f 919a124a 1dfbba19 be004edc**, Maka selanjutnya menerapkan metode DSA .

5. Pembangkit Sepasang Kunci

a. p dan q adalah bilangan prima, dimana p dengan panjang L bit atau $512 \leq L \leq 1024$ dengan kelipatan 64, $(p-1)\ Mod\ q = 0$.
$\boldsymbol{p} = 38977$ dan $\boldsymbol{q} = 2436$

$(38977 - 1)\ Mod\ 2436 = 0$

b. Menghitung parameter $g = h^{(p-1)/q}\ Mod\ p$ yang bersifat *public*, dimana $1 < h < p - 1$ dan $h^{(p-1)/q}\ Mod\ p > 1$.
$h = 100$
$\boldsymbol{g} = 100^{(38977\ -1)/2436}\ Mod\ 38977 = 4820$

c. Menentukan nilai sembarang untuk parameter x atau kunci *private* yang merupakan bilangan bulat, dimana $x < q$.
$\boldsymbol{x} = \mathbf{203}$ **(Sebagai kunci *Private*)**

d. Menghitung nilai pada kunci *public* $y = g^x\ Mod\ p$.
$y = 4820^{203}\ Mod\ 38977$
$\boldsymbol{y} = \mathbf{20297}$ **(Sebagai kunci *Public*)**

6. Proses Enkripsi (Tanda Tangan Pada Dokumen)
Input : Pesan (M) dan kunci *private* (x)
Output : Pesan (M) dan tanda tangan (r, s)

a. Ubah nilai *hash* dari **206 (**hexadesimal**)** kedalam bentuk bilangan bulat (desimal), yaitu sebagai berikut :
H(m) = 4afa8f9e90756f0f919a124a1dfbba19be004edc
H(m) = **428053014354117578508610761906229859966888464092**

b. Menentukan bilangan acak $k < q$
$\boldsymbol{k}$ = 571

c. Menghitung r dan s tanda-tanda dari pesan, yaitu sebagai berikut :
$\boldsymbol{r} = (g^k \bmod p) \bmod q$
$= (4820^{571} \bmod 38977\ ) \bmod 2436$
$= \mathbf{67}$
$\boldsymbol{s} = (k^{-1}\ (\mathrm{H(m)} + x * r)) \bmod q.\ k^{-1}$ merupakan invers k mod q.
$\boldsymbol{k^{-1}}$ = **1843**
$\boldsymbol{s} = \big(1843(428053014354117578508610761906229859966888464092 + 203 * 67)\big) \bmod 2436$ = **1307**

d. Pesan (M) dapat dikirim beserta tanda tangan r dan s.
Yaitu data ***e-invoice*** beserta **digital signature 671307**

7. Proses Dekripsi (Validasi pada Dokumen)
Teorema 1 : (Pembuktian *v = r'*)
Jika *M' = M*, *r' = r*, dan *s' = s* pada verifikasi tandatangan, maka *v = r'*.

a. $\boldsymbol{s^{-1}}$ = Invers s mod q
$=$ Invers 1307 mod 2436 $= \mathbf{479}$

b. $\boldsymbol{w} = s^{-1} \bmod q$
$= 479 \bmod 2436 = \mathbf{479}$

c. **u1** = (H(m) * w) mod q
= (428053014354117578508610761906229859966888464092 * 479) mod 2436 = **1992**

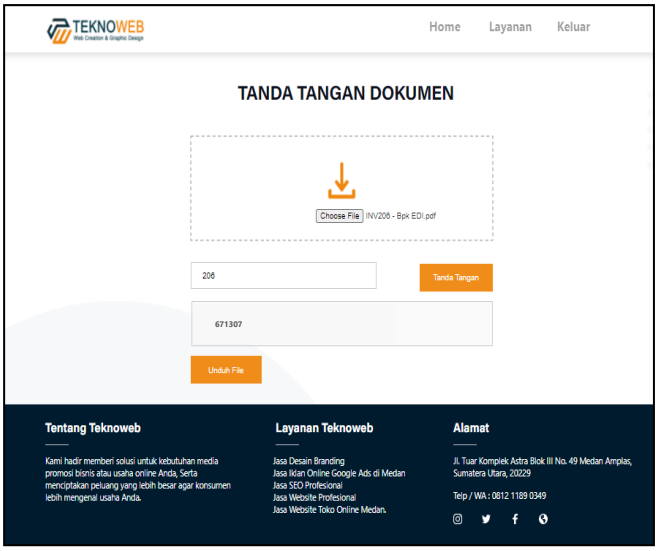
d. **u2** = (r * w) mod q
= (67 * 479) mod 2436 = **425**

e. **v** = $((g^{u1} * y^{u2}) \bmod p) \bmod q$)
= $((4820^{1992} * 20297^{425}) \bmod 38977\ ) \bmod 2436)$
= **67**

Karena v = r', maka tanda tangan dinyatakan asli.

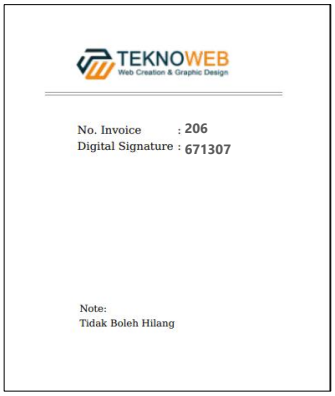
3.4 Pengujian

Pada tahapan membuat tanda tangan pada dokumen, langkah yang harus dilakukan adalah memilih dokumen e-*invoice* dalam bentuk PDF dan membuat tanda tangan yang berfungsi sebagai data *sign* untuk keabsahan dari dokumen e-*invoice* yang akan dikirim dengan cara menggunakan kunci privat yang telah di bentuk (*generate)* secara sistematis dan dinamis oleh sistem. Tahap tanda tangan ini dapat dilihat pada gambar berikut.



Gambar 2. Pembentukan Tanda Tangan Dokumen *E-Invoice*

Setelah dokumen ditanda tangani maka admin akan mendapatkan *digital signature* yang dapat diunduh berupa PDF. Adapun tampilan PDF pada *digital signature* yaitu sebagai berikut :

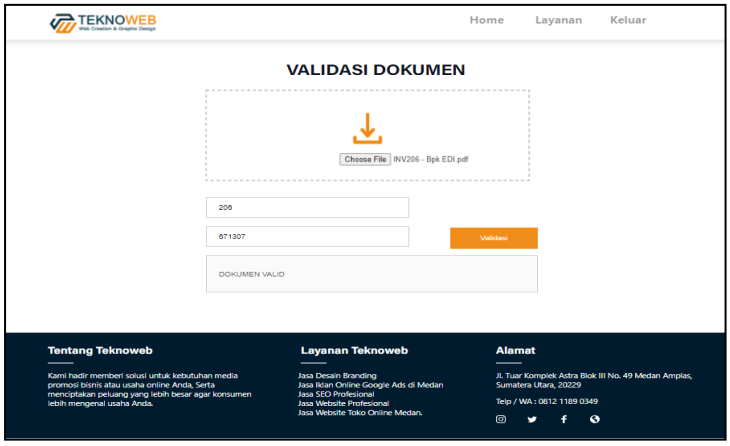


Gambar 3. *Digital Signature*

Untuk melakukan pemeriksaan keabsahan dokumen *e-invoice*, maka dilakukan proses verifikasi oleh pihak admin sebelum adanya tindak lanjut dari keluhan (*complain*) pelanggan. Proses verifikasi ini dilakukan untuk mengetahui bahwa dokumen masih terjaga integritas nya atau tidak. Langkah ini dilakukan dengan cara memilih dokumen *e-invoice*, memasukan *no.invoice*, *digital signature* serta menggunakan kunci publik yang telah dibangkitkan atau dibentuk (*generate)* secara sistematis dan dinamis. Pada tahap ini penulis melakukan uji coba untuk memvalidasi dokumen *e-invoice* yang berbeda yaitu adanya perubahan pada jumlah nominal pembayaran *(amount)*. Adapun perbedaan pada dokumen *e-invoice* dapat dilihat sebagai berikut :

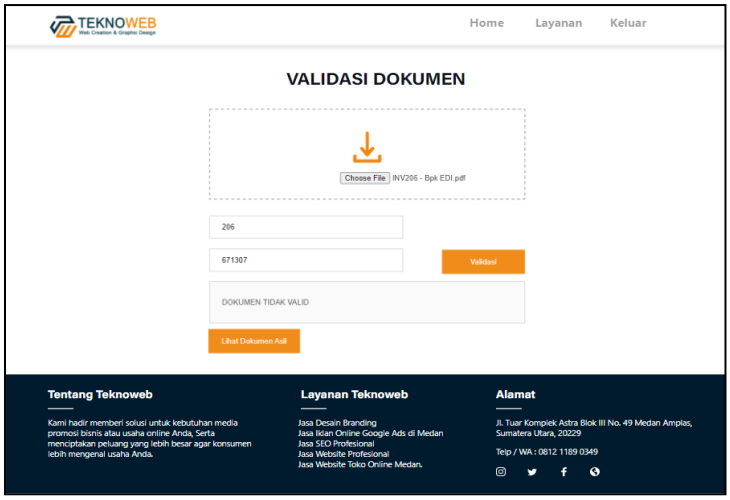
Gambar 4. Perbandingan Dokumen *E-Invoice*

Ketika dilakukan proses validasi dokumen (dekripsi) pada dokumen *e-invoice* yang asli, maka sistem akan memvalidasi keabsahan dokumen dengan memberikan informasi "Dokumen Valid". Adapun hasilnya sebagai berikut :



Gambar 5. Dokumen Valid

Jika dokumen *e-invoice* yang sudah mengalami perubahan pada *amount* dilakukan proses validasi dokumen (dekripsi), maka sistem memberikan informasi "Dokumen Tidak Valid" dan admin dapat melakukan perbandingan dengan dokumen *e-invoice* asli. Adapun hasilnya sebagai berikut :



Gambar 6. Dokumen Tidak Valid

Berdasarkan hasil pengujian yang dilakukan adanya perbedaan dokumen *e-invoice* yang cukup signifikan, hal ini terjadi karena dokumen *e-invoice* yang telah diberikan tanda tangan digital sudah dilakukan fungsi *hash* sehingga nilai *hash* yang dihasilkan dapat membuat sistem akan mendeteksi keabsahan dokumen *e-invoice.* Hal ini merupakan salah satu karakter dari metode yang tidak dapat menghasilkan nilai *hash* yang sama pada dokumen *e-invoice* yang berbeda. Oleh sebab itu *digital signature* yang diperoleh merupakan *digital signature with message recovery* yaitu menyajikan unit data *signed* dan *digital signature* pada file yang berbeda.

4. KESIMPULAN

Adapun kesimpulan dari penelitian ini yaitu sebagai berikut :

1. Berdasarkan pengujian proses pembangkitan sepasang kunci dilakukan secara sistematis dan dinamis tanpa dilakukan penginputan secara manual.
2. Berdasarkan pengujian sistem dapat melakukan proses membuat tanda tangan pada dokumen *e-invoice*.
3. Berdasarkan pengujian sistem dapat melakukan verifikasi sehingga mampu mendeteksi keabsahan dari dokumen *e-invoice*.
4. Berdasarkan pengujian sistem dapat diimplementasikan dengan menerapkan algoritma SHA dan DSA.
5. Berdasarkan hasil penelitian keamanan sistem dapat ditingkatkan dengan modifikasi kedua metode sehingga tidak mudah untuk mendekripsikan dokumen *e-invoice* yang berbeda untuk tetap menghasilkan informasi yang sama.
6. Berdasarkan hasil penelitian sistem *digital signature* dapat membantu untuk mengatasi masalah yang terjadi pada Teknoweb Indonesia.
7. Berdasarkan hasil penelitian sistem *digital signature* dapat mendeteksi adanya perubahan pada seluruh isi dokumen *e-invoice*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada program studi S1 Sistem Informasi STMIK Triguna Dharma yang telah memberikan dukungan dalam penyelesaian tulisan ini.

REFERENSI

[1] R. Perdana, D. Anbiya, And A. Grahitandaru, "Penerapan Tanda Tangan Digital Pada Gambar Formulir C1.Plano-Kwk Di Pilkada Sulawesi Selatan," *Teknol. Inf. Dan Ilmu Komput.*, Vol. 6, No. 5, Pp. 475–484, 2019, Doi: 10.25126/Jtiik.201961471.

[2] R. Munir, Kriptografi, 2nd Ed. Yogyakarta: Informatika Bandung, 2019.

[3] J. Simarmata, S. Sriadhi, And R. Rahim, Kriptografi, 1st Ed. Yogyakarta: Andi, 2019.

[4] F. Nurhasanah And R. Sulaiman, "Pembuatan Tanda Tangan Digital Menggunakan Digital Signature Algorithm," *J. Mipa Univ. Negeri Surabaya*, 2011.

[5] R. Ndaumanu, "Perancangan Sistem Informasi Persediaan Obat Pada Apotek Rumah Sakit Menggunakan Metode Spiral," *J. Komput. Dan Inform.*, Vol. 8, No. 1, Pp. 18–27, 2020, Doi: 10.35508/Jicon.V8i1.2187.

[6] A. Cahyono, "Aplikasi Digital Signature Untuk Pengaman E-Document Di Pg. Pesantren Baru Menggunakan Algoritma Dsa," *Artik. Ilm.*, Vol. 2, Pp. 227–249, 2018.

[7] H. Wibowo, N. Cahyani, And V. Suryani, "Implementasi Digital Signature Algorithm (Dsa) Dalam Keamanan Sms Pada Mobile Device," *J. Ilm.*, Pp. 1–7, 2010.

[8] "Schneier On Security," *Schneier.Com*, 2015. Http://Www.Schneier.Com/Blog/Archives/2005/02/Sha_Ibroken.Html (Accessed Jan. 02, 2021).

[9] R. Wicaksono, "Membuat-Sha1-Collision-File," *Ilmuhacking.Com*, 2017. Http://Www.Ilmuhacking.Com/Cryptography/Membuat-Sha1-Collision-File/ (Accessed Jan. 02, 2021).

[10] M. Ihwani, "Model Keamanan Informasi Berbasis Digital Signature," *Cess (Journal Comput. Eng. Syst. Sci.*, Vol. 1, No. 1, Pp. 15–20, 2016

[11] P. Chyan, "Penerapan Sistem Kriptografi Enkripsi Jamak Dan Tanda Tangan Digital Dalam Mendukung Keamanan Informasi," *J. Temat.*, Vol. 6, No. 1, Pp. 39–46, 2018
[12] Teknoweb Indonesia, Data *E-Invoice*.
[13] R. Munir, "Digital Signature Standard," *Encycl. Cryptogr. Secur.*, Pp. 347–347, 2011, Doi: 10.1007/978-1-4419-5906-5_145.

BIBLIOGRAFI PENULIS

Radika Maidinda Lahir pada tahun 1999 di Kualasimpang, NAD. Saat ini sedang menempuh studi Sistem Informasi di STMIK Triguna Dharma. Sejak tahun 2019 hingga saat ini bekerja sebagai Designer di Uniqa. Menjabat sebagai pengurus di organisasi daerah yaitu Himpunan Mahasiswa Tamiang (HMT) pada tahun 2020. Aktif sebagai relawan kemanusiaan serta penerima Beasiswa di lembaga amil zakat yaitu DT Peduli dan Dompet Dhuafa Waspada, juga sebagai penerima Beasiswa daerah tahun 2018-2020. Pada tahun 2018 menjadi salah satu delegasi Indonesia dalam kompetisi NetRiders Asia Pasific Japan yang diselenggarakan oleh Cisco. Menyelesaikan Program Kreativitas Mahasiswa (PKM-P) bersama rekan lainya pada tahun 2020.

Dr. Dicky Nofriansyah., S.Kom., M.Kom Lahir pada tahun 1989 di Medan. Menyelesaikan pendidikan Strata 1 di STMIK Budi Dharma, Strata 2 di Universitas Putra Indonesia "YPTK" Padang, dan Strata 3 di Universitas Negeri Padang. Saat ini merupakan Dosen tetap STMIK Triguna Dharma, Medan. Mengampu matakuliah Data Minning, Metodologi Penelitian, dan Sistem Pendukung Keputusan. Pada 2020 telah menyelesaikan penelitian serta pengabdian kepada masyarakat dengan penyandang dana pribadi. Menulis buku dengan judul "Sistem Pendukung Keputusan : metode dan implementasi" serta menulis artikel Ilmiah untuk publikasi ditahun 2020. Menjadi salah satu Dosen terbaik di STMIK Triguna Dharma tahun 2019 juga aktif mengikuti pertemuan Ilmiah atau Seminar "International Conference : NICCT" di Universitas Nomensen.

Masyuni Hutasuhut., S.Kom., M.Kom Lahir pada tahun 1992 di Toboton. Saat ini merupakan Dosen tetap STMIK Triguna Dharma, Medan. Mengampu matakuliah E-Bisnis, Data Warehouse dan Data Minning. Pada 2019 berhasil menyelesaikan penelitian serta pengabdian kepada masyarakat dengan penyandang dana STMIK Triguna Dharma. Menyelesaikan penelitian dengan penyandang dana Dikti di tahun 2020.