Jurnal CyberTech

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

Implementasi *Digital Signature* Pada *E-Invoice* Di Uniqa Digital Invitation Menggunakan Algoritma SHA-256 (*Secure Hash Algorithm-256*) Dan RSA (*Rivest Shamir Adleman*)

Fauzi Maulana Rangkuti *, Nurcahyo Budi Nugroho **, Zaimah Panjaitan **

* 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:
Kriptografi asimetris
Digital signature
Tanda tangan digital
SHA-256 *(Secure Hash Algorithm-256)*
RSA *(Rivest Shamir Adleman)*
Fungsi *hash*
Enkripsi

ABSTRACT

Di era pesatnya perkembangan teknologi ini, digitalisasi pada berkas maupun dokumen merupakan sebuah kebutuhan. Hal ini akan mempermudah pihak instansi untuk mendistribusikan dokumen *e-invoice* melalui berbagai media komunikasi elektronik. Aspek keamanan integritas data dan autentikasi merupakan hal yang sangat penting dalam menjaga keamanan dan kerahasiaan data dari pihak tidak bertanggung jawab yang turut andil dalam memanfaatkan data untuk kepentingan pribadi.Salah satu tindakan dalam pencegahan yaitu dengan memanfaatkan ilmu kriptografi yang mengubah pesan menjadi kode rahasia yang sulit dipahami. Sehingga tuntutan penerapan aspek keamanan autentikasi dan integritas pada data menjadi terpenuhi.SHA-256 (*Secure Hash Algorithm-256*) dan RSA (*Rivest Shamir Adleman*) merupakan salah satu jenis kriptografi yang dapat diterapkan. Penerapan algoritma SHA-256 akan melindungi pesan dengan melakukan proses *hash* dari rangkuman dokumen *e-invoice* sehingga menghasilkan nilai *hash* yang berbeda-beda dalam setiap e-*invoice*. Sedangkan algoritma RSA memaksimalkan nilai hash untuk dilakukan proses penyandian dengan menghasilkan kode yang rumit demi memberikan jaminan autentikasi pengirim maupun penerima dokumen.

Corresponding Author:
Nama : Fauzi Maulana Rangkuti
Sistem Informasi
STMIK Triguna Dharma
Email: fauzimawlana@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[1]. Invoice adalah salah satu dokumen penting dalam menjalankan bisnis yang berfungsi sebagai perekam transaksi yang dilakukan. Salah satu keunggulan berbisnis di dunia maya adalah dengan dilakukannya transaksi perdagangan kapan saja dan dimana saja tanpa adanya tatap muka secara fisik antara penjual dan pembeli. Hal ini kerapa menjadi permasalahan tersendiri, terutama yang berhubungan dengan masalah autentikasi, yaitu dengan meyakinkan penjual bahwa yang membeli atau menggunakan produk dan jasanya adalah orang yang sesungguhnya, atau menyakinkan penjual bahwa informasi yang dikirimkan oleh penjual tidak jatuh ke tangan oknum yang tidak berhak selain pembeli yang bersangkutan, serta transaksi yang dilakukan sah secara hukum karena tidak adanya pihak penipuan dari pembeli. Keuntungan dengan distribusi invoice melalui media komunikasi digital adalah dapat memangkas pemborosan dalam prosesnya seperti biaya yang lebih murah tanpa adanya pembelian kertas, dan dapat memangkas waktu yang lebih cepat[2].

Terdapat beberapa aspek yang harus dipenuhi dalam keamana data dan informasi, yaitu aspek kerahasiaan (*confidentially*), ketersediaan (*availability*), keutuhan data (*integrity*), penyedia /penerima informasi (*authentication*), dan anti penyangkalan (*non-repudiation*)[3]. Sejalan dengan pentingnya aspek keamanan sistem informasi, kriteria-kriteria tersebut harus terdapat dan terpenuhi pada *e-invoice* terutama dalam kriteria konsep integritas, autentikasi, dan anti penyangkalan. Untuk itu perlu diterapkannya suatu metode pengamanan data sebagai bukti dari keaslian dan integritas dari *e-invoice* yang diharapkan agar keamanan pada data tersebut bisa lebih ditingkatkan lagi mengingat seiring dengan pesatnya perkembangan teknologi. Semakin majunya perkembangan teknologi mengakibatkan sistem semakin rentan terserang oleh virus atau hal-hal yang dapat merusak kinerja sistem sehingga memunculkan resiko mudahnya kebocoran data. Tanda tangan digital merupakan salah satu dari banyaknya perkembangan dan inovasi dalam dunia keamanan data maupun jaringan. Dengan autentikasi data dan integritas data, penerima pesan dengan media komunikasi digital dapat lebih mempercayai pengirim pesan.

Kriptografi merupakan ilmu yang mempelajari bagaimana agar data aman, tidak bisa dibaca oleh pihak yang tidak berhak atas data tersebut, sehingga menimbulkan banyak kerugian[4]. Kriptografi bekerja dengan cara menyamarkan data asli dengan suatu metode kriptografi tertentu sehingga menghasilkan karakte acak yang unik guna memenuhi tujuan utama kriptografi yaitu *authentication*, *integrity*, *non-repudiation*, *confidentially*, *availability*[3].

Tanda tangan digital merupakan salah satu fasilitas yang disediakan oleh kriptografi. Tanda tangan digital sangat bergantung pada dokumen atau data yang di tanda tanganinya, sehingga hasil tanda tangan digital sepenuhnya berbeda dengan tiap-tiap dokumen[5]. Ada berbagai macam algoritma kriptografi yang dapat digunakan dan di implementasikan kedalam sistem, tiap-tiap algoritma memiliki kelebihan dan kekurangan yang berbeda-beda. Pembuatan tanda tangan digital dimulai dengan proses mendapatkan data kunci yang bertindak sebagai *plaintext*, kemudian data tersebut dilakukan proses perhitungan komputasi fungsi *hash* dengan algoritma SHA-256 untuk mendapatkan *message digest*, lalu hasil fungsi *hash* tersebut di enkripsi dengan algoritma RSA yang merupakan algoritma kriptigrafi asimetris yang menggunakan 2 kunci berbeda setiap proses enkripsi dan dekripsinya.

Kedepanya hal baru yang ada di Uniqa Digital Invitation adalah sistem *digital signature* berbasis web yang mengadopsi Algoritma SHA-256 (*Secure Hash Algoritm-256*) dan RSA (*Rivest Shamir Adleman*) yang diharapkan mampu menyelesaikan masalah terkait keaslian dokumen *e-invoice*. Berdasarkan kondisi tersebut maka diangkat judul penelitian yaitu: "**Implementasi *Digital Signature* Pada *E-Invoice* Di Uniqa Digital Invitation Menggunakan Algoritma SHA-256 (*Secure Hash Algorithm-256*) Dan RSA (*Rivest Shamir Adleman*)**".

2. METODE PENELITIAN

Metode pengembangan yang diadopsi adalah model *waterfall*. Model *waterfall* adalah metode pengembangan sistem yang memungkinkan pembuatan sistem dilakukan secara terstruktur, berurutan dan sistematis yang mengusulkan pendekatan sistematik dan sekuensial yang dimulai pada tingkay tahapan analisis hingga pemeliharaan[6].

Terdapat 5 tahapan dalam model pengembangan sistem waterfalll, antara lain :

1. *Requirement* (Analisa Kebutuhan)
   Komunikasi sekaligus identifikasi ke pihak administrasi guna mengumpulkan data, menganalisa masalah dan mengeluarkan solusi sesuai kebutuhan.
2. Desain Sistem
   Proses yang berfokus pada pemodelan sistem, perancangan struktur data, dan perancangan interface sistem.
3. Implementasi
   Proses ini fokus pada pembuatan atau pembangunan aplikasi yang dilakukan oleh *programmer* dengan kode bahasa pemrograman tertentu yang mendukuung dalam pembangunan sistem.
4. *Integrasi dan pengujian*
   Kegiatan yang dilakukan untuk menguji sistem yang telah dibangun. Sistem yang diuji terkait interface, database sekaligus fungsi syntax-sytax pada pemroses.
5. *Pemeliharaan*
   Implementasi sistem ke instansi sebagai mempermudah dan solusi permasalahan.

3. ANALISA DAN HASIL

Kombinasi algoritma kriptografi *modern* yang salah satu fungsinya sebagai validasi autentikasi sebuah dokumen digital yang dikirim melalui media komunikasi digital. Dengan melakukan fungsi *hash* untuk menghasilkan *message digest*, lalu *message digest* di enkripsi dengan algoritma kriptografi asimetris yang memiliki 2 buah kunci distribusi. Hal ini dilakukan sebagai langkah pengamanan data dan mencegah modifikasi atau fabrikasi data yang dilakukan pihak yang tidak bertanggung jawab.

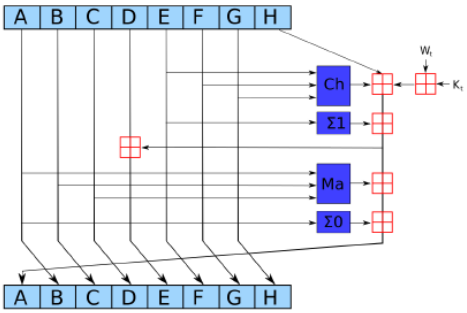
3.1. SHA-256 (*Secure Hash Algorithm-256)*

SHA-256 (*Secure Hash Algorithm-256)* adalah sebuah kriptografi fungsi hash satu arah yang dirancang oleh National Security Agency (NSA) dan dipublikasikan oleh National Institute of Standards and Technology (NIST) sebagai Federal Information Processing Standard (FIPS) oleh U.S. Secure Hash Standard menetapkan 4 algoritma keamanan fungsi hash yaitu SHA-1, SHA-256, SHA-384 dan SHA 521[7][8].

Tabel 1. Macam-macam SHA

Algoritma		**Ukuran Pesan (bit)**	**Ukuran Blok (bit)**	**Ukuran Kata (bit)**	**Ukuran *message digest* (bit)**
SHA-1	SHA-1	$<2^{64}$	512	32	160
SHA-2	SHA-256	$<2^{64}$	512	32	256
	SHA-384	$<2^{128}$	1024	64	384
	SHA-521	$<2^{128}$	1024	64	512

Nilai fungsi hash digunakan untuk autentikasi pesan dengan menghitung nilai kode hash sebagai fungsi bit. Berdasarkan Secure Hash Signature Standard, apabila pesan yang panjangnya lebih pendek dari 2^{64} bit, maka yang mengoperasikan hashnya adalah 512 bit dalam kelompok dan menjadi message digest 256-bit[8][9].



Gambar 1. Jalur Komputasi SHA-256

3.2. RSA (*Rivest Shamir Adleman)*

Shifford Cocks, James H. Ellis dan Malcolm Williamson merupakan sekelompok ahli matematika yang menemukan algoritma kriptografi asimetris. Pada tahun 1976 algoritma kunci asimetris pertama kali dipublikasikan oleh Whitfirld Diffie dan Martin Hellman. Pada tahun 1977 Ron Rivest, Adi Shamir dan Leonard Adleman dari Massachussets Institute of Technology menemukan kembali ide dari Clifford Cocks terkait algoritma kriptografi asimetris lalu mempublikasikannya pada tahun 1978[10].

Algoritma kriptografi berkonsep asimetris yang terdiri dari kunci publik dan kunci private. Algoritma RSA menggunakan kunci publik ketika akan melakukan proses enkripsi pesan dari plainteks menjadi cipherteks, dan menggunakan kunci privat ketika akan melakuka proses dekripsi dari cipherteks menjadi plainteks. RSA ini memanfaatkan 2 bilangan prima dalam proses pembangkitan kunci publik dan kunci privat[11].

Terdapat 3 proses tahapan dalam RSA, yaitu[4]:

1. Proses pembangkitan kunci
2. Proses enkripsi
3. Proses dekripsi

3.3 Penerapan Dengan Metode

Berikut ini adalah data yang digunakan sebagai sampel dalam penelitian yaitu

Tabel 2. Data Sampel

No. Invoice					
0	0	1	0	1	1

Data sampel selanjutnya akan dilakukan proses hashing dengan tahapan yang terdapat pada algoritma SHA-256, yaitu:

1. *Padding Bit* (Penambahan Pesan)

   Menambkan bit pengganjal (padding) dengan angka 0 sejumlah dengan k

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

   Berdasarkan data sampel yang digunakan dengan No.Invoice “001011” setelah di konversikan memiliki nilai ASCII “”memiliki panjang pesan 17 * 8 = 136.

   k + 136 + 1 ≡ 448 mod 512

   k + 137 ≡ 448 mod 512

   k ≡ 448 -137 mod 512

   k ≡ 311 mod 512

   Maka banyaknya bit pengganjal 0 yang di tambahkan sebanyak 311 bit.

Tabel 3. Padding Bit

0110100	00111000	00100000	00110100	00111000	00100000	00110100	00111001
00100000	00110100	00111000	00100000	00110100	00111001	00100000	00110100
00111001	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

2. Penambahan Panjang *Append*

Tambahkan jumlah panjang pesan pada akhir pesan yang dipadding sebanyak 64 bit. Panjang pesan dari "001011" adalah 136, maka biner dari 136 adalah 10001000.

Tabel 4. Panjang Append

0110100	00111000	00100000	00110100	00111000	00100000	00110100	00111001
00100000	00110100	00111000	00100000	00110100	00111001	00100000	00110100
00111001	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	10001000

3. *Parsing* (Penguraian pesan)

Melakukan penguraian pesan dengan membagi setiap blok 512 bit menjadi 16 blok, tiap blok berukuran 32 bit.

Tabel 4. Hasil *Parsing* Pesan

Blok	Biner	Blok	Biner
M_0	00110100001110000010000000110100	M_8	00000000000000000000000000000000
M_1	00111000001000000011010000111001	M_9	00000000000000000000000000000000
M_2	00100000001101000011100000100000	M_{10}	00000000000000000000000000000000
M_3	00110100001110010010000000110100	M_{11}	00000000000000000000000000000000
M_4	00111001100000000000000000000000	M_{12}	00000000000000000000000000000000
M_5	00000000000000000000000000000000	M_{13}	00000000000000000000000000000000
M_6	00000000000000000000000000000000	M_{14}	00000000000000000000000000000000
M_7	00000000000000000000000000000000	M_{15}	00000000000000000000000010001000

4. Inisialisasi Nilai *Hash*

Tabel 5. Inisialisasi Nilai *Hash*

Variabel	Nilai Hash	Variabel	Nilai Hash
$H_0^{(0)}$	6A09E667	$H_4^{(0)}$	510E527F
$H_1^{(0)}$	BB67EA85	$H_5^{(0)}$	9B05688C
$H_2^{(0)}$	3C6EF372	$H_6^{(0)}$	1F83D9AB
$H_3^{(0)}$	A54FF53A	$H_7^{(0)}$	5BE0CD19

5. Penjadwalan Pesan

Pada tahap ini diawali dengan mengubah atau konversi setiap blok pesan menjadi bilangan heksadesimal dengan ketentuan sebagai berikut :

$$Wt = \begin{cases} M_t^{(t)} & 0 \le t \le 15 \\ \sigma_1^{(256)}(Wi-2)+Wi-7+\sigma_0^{(256)}(Wi-15)+Wi-16 & 16 \le t \le 63 \end{cases} \quad \text{.....[3.2]}$$

Tabel 6. Penjadwalan Pesan

W_0	34382034	W_{16}	ACB2291B	W_{32}	43F6051C	W_{48}	B08C74A3
W_1	38203439	W_{17}	82C41BB2	W_{33}	867cd7f6	W_{49}	C347A312
W_2	20343820	W_{18}	1ECF931C	W_{34}	D1532741	W_{50}	9B9F5519
W_3	34392034	W_{19}	060ED1FF	W_{35}	CE48EA8D	W_{51}	FB291692
W_4	39800000	W_{20}	3BEA3F5A	W_{36}	B138DC9F	W_{52}	3790C59C
W_5	00000000	W_{21}	B2C1E072	W_{37}	E430BEE9	W_{53}	B027ED2A
W_6	00000000	W_{22}	5848A007	W_{38}	C82E9ACA	W_{54}	8E137571
W_7	00000000	W_{23}	107E62A2	W_{39}	4B3CA808	W_{55}	0F5A61E7
W_8	00000000	W_{24}	14A3FA29	W_{40}	423AC2D8	W_{56}	9F6563D1
W_9	00000000	W_{25}	1BD0E918	W_{41}	5ECE0327	W_{57}	A1B899E4
W_{10}	00000000	W_{26}	8863523A	W_{42}	AF4CEB02	W_{58}	CA96BB11
W_{11}	00000000	W_{27}	A5943A02	W_{43}	DC6BA178	W_{59}	037DE0DE
W_{12}	00000000	W_{28}	7639EE5B	W_{44}	E8A1BD7E	W_{60}	9AF61733
W_{13}	00000000	W_{29}	F2B1037D	W_{45}	415ABFE8	W_{61}	EC9EFF3E
W_{14}	00000000	W_{30}	9B91BE4E	W_{46}	F82A0926	W_{62}	B73DF827
W_{15}	00000088	W_{31}	1B6DE5A9	W_{47}	178062FE	W_{63}	A381CD90

6. Inisialisasi Variabel Kerja Dan Konstanta

Setiap variabel kerja *a*,*b*,*c*,*d*,e,*f*,*g* dan *h* diambil dari nilai inisialisasi nilai *hash*.

a = $H_0^{(0}$ = 6A09E667
b = $H_1^{(0}$ = BB67EA85
c = $H_2^{(0}$ = 3C6EF372
d = $H_3^{(0}$ = A54FF53A
e = $H_4^{(0}$ = 510E527F
f = $H_5^{(0}$ = 9B05688C
g = $H_6^{(0}$ = 1F83D9AB
h = $H_7^{(0}$ = 5BE0CD19

Tabel 7. Nilai Konstanta

428A2F98	71374491	B5C0FBCF	E9B5DBA5	3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3	72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC	2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7	C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13	650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3	D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5	391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208	90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

7. Komputasi Fungsi *Hash*

Selanjutnya dilakukan penyelesaian proses komputasi fungsi hash dari $t = 0$ sampai $t = 63$.

Tabel 8. Variabel Nilai Hash

Variabel	Nilai Hash	Variabel	Nilai Hash
a	6A09E667	*e*	510E527F
b	BB67EA85	*f*	9B05688C
c	3C6EF372	*g*	1F83D9AB
d	A54FF53A	*h*	5BE0CD19

Tabel 9. Hasil Komputasi Nilai Hash

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
Init	6A09E667	BB67AE85	3C6EF372	A54FF53A	510E527F	9B05688C	1F83D9AB	5BE0CD19
t=0	FC088946	6A09E667	BB67AE85	3C6EF372	98c7e39b	510E527F	9B05688C	1F83D9AB
t=1	F4afbc47	FC088946	6A09E667	BB67AE85	6013278d	98c7e39b	510E527F	9B05688C
t=2	D58763AC	F4afbc47	FC088946	6A09E667	F300102C	6013278d	98c7e39b	510E527F
t=3	83FFF8CA	D58763AC	F4afbc47	FC088946	C3F06E65	F300102C	6013278d	98c7e39b
t=4	C79AF987	83FFF8CA	D58763AC	F4afbc47	54AABE3D	C3F06E65	F300102C	6013278d
t=5	232A1FDE	C79AF987	83FFF8CA	D58763AC	F9FB44EF	54AABE3D	C3F06E65	F300102C
t=6	19C8467D	232A1FDE	C79AF987	83FFF8CA	8D2BCA30	F9FB44EF	54AABE3D	C3F06E65
t=7	B285DF17	19C8467D	232A1FDE	C79AF987	DE784C48	8D2BCA30	F9FB44EF	54AABE3D
t=8	A78AD853	B285DF17	19C8467D	232A1FDE	383CB55A	DE784C48	8D2BCA30	F9FB44EF
t=9	3B5EE9EA	A78AD853	B285DF17	19C8467D	AAE16695	383CB55A	DE784C48	8D2BCA30
t=10	8FF3B15C	3B5EE9EA	A78AD853	B285DF17	3C0B5626	AAE16695	383CB55A	DE784C48
t=11	CD104BE2	8FF3B15C	3B5EE9EA	A78AD853	67DD4FAA	3C0B5626	AAE16695	383CB55A
t=12	4669AFE1	CD104BE2	8FF3B15C	3B5EE9EA	B123ACFC	67DD4FAA	3C0B5626	AAE16695
t=13	ACEFD567	4669AFE1	CD104BE2	8FF3B15C	90AD35C5	B123ACFC	67DD4FAA	3C0B5626
t=14	937FEBEB	ACEFD567	4669AFE1	CD104BE2	57B6C0D0	90AD35C5	B123ACFC	67DD4FAA
t=15	24EF7385	937FEBEB	ACEFD567	4669AFE1	2763003C	57B6C0D0	90AD35C5	B123ACFC
t=16	C6365153	24EF7385	937FEBEB	ACEFD567	FED31E44	2763003C	57B6C0D0	90AD35C5
t=17	5423F629	C6365153	24EF7385	937FEBEB	D83D6CCF	FED31E44	2763003C	57B6C0D0
t=18	ED8B00DE	5423F629	C6365153	24EF7385	D1488003	D83D6CCF	FED31E44	2763003C
t=19	5152C57A	ED8B00DE	5423F629	C6365153	208810CB	D1488003	D83D6CCF	FED31E44
t=20	FC69E1EC	5152C57A	ED8B00DE	5423F629	79089C38	208810CB	D1488003	D83D6CCF
t=21	EDA6EC18	FC69E1EC	5152C57A	ED8B00DE	AC84F77E	79089C38	208810CB	D1488003
t=22	C2E835B3	EDA6EC18	FC69E1EC	5152C5 7A	F2F19B92	AC84F77E	79089C38	208810CB
t=23	7DBC9F83	C2E835B3	EDA6EC18	FC69E1EC	E433AA1F	F2F19B92	AC84F77E	79089C38
t=24	F31193D9	7DBC9F83	C2E835B3	EDA6EC18	30C3F040	E433AA1F	F2F19B92	AC84F77E
t=25	53F5BD85	F31193D9	7DBC9F83	C2E835B3	A99F7654	30C3F040	E433AA1F	F2F19B92
t=26	C95BF746	53F5BD85	F31193D9	7DBC9F83	AA6CA7F5	A99F7654	30C3F040	E433AA1F
t=27	5B0F440A	C95BF746	53F5BD85	F31193D9	9DBC7F9D	AA6CA7F5	A99F7654	30C3F040
t=28	58E8B03C	5B0F440A	C95BF746	53F5BD85	65192EFB	9DBC7F9D	AA6CA7F5	A99F7654
t=29	4E488318	58E8B03C	5B0F440A	C95BF746	13DA3266	65192EFB	9DBC7F9D	AA6CA7F5
t=30	973E0DE8	4E488318	58E8B03C	5B0F440A	DCF5515B	13DA3266	65192EFB	9DBC7F9D
t=31	4EF1D19E	973E0DE8	4E488318	58E8B03C	98DBFB9A	DCF5515B	13DA3266	65192EFB
t=32	A0F5DD61	4EF1D19E	973E0DE8	4E488318	3A4DFA33	98DBFB9A	DCF5515B	13DA3266
t=33	2D1B6315	A0F5DD61	4EF1D19E	973E0DE8	A02811D0	3A4DFA33	98DBFB9A	DCF5515B
t=34	0C42E386	2D1B6315	A0F5DD61	4EF1D19E	382B3BAF	A02811D0	3A4DFA33	98DBFB9A
t=35	7A66E8AB	0C42E386	2D1B6315	A0F5DD61	0856147D	382B3BAF	A02811D0	3A4DFA33
t=36	75E97065	7A66E8AB	0C42E386	2D1B6315	EA3BA54B	0856147D	382B3BAF	A02811D0
t=37	A5CF4DA1	75E97065	7A66E8AB	0C42E386	DAF7698A	EA3BA54B	0856147D	382B3BAF
t=38	5D0BFF72	A5CF4DA1	75E97065	7A66E8AB	BA16FED2	DAF7698A	EA3BA54B	0856147D
t=39	14FE1F66	5D0BFF72	A5CF4DA1	75E97065	D66BABB9	BA16FED2	DAF7698A	EA3BA54B
t=40	ABC61262	14FE1F66	5D0BFF72	A5CF4DA1	856D6AEB	D66BABB9	BA16FED2	DAF7698A
t=41	25A76EE2	ABC61262	14FE1F66	5D0BFF72	8BFB4D1A	856D6AEB	D66BABB9	BA16FED2
t=42	BF9B1B7E	25A76EE2	ABC61262	14FE1F66	92FD7E79	8BFB4D1A	856D6AEB	D66BABB9
t=43	22095D68	BF9B1B7E	25A76EE2	ABC61262	6F01A173	92FD7E79	8BFB4D1A	856D6AEB
t=44	13E12303	22095D68	BF9B1B7E	25A76EE2	D164EE63	6F01A173	92FD7E79	8BFB4D1A
t=45	64C7495D	13E12303	22095D68	BF9B1B7E	FE78C14F	D164EE63	6F01A173	92FD7E79
t=46	ACBEF4C0	64C7495D	13E12303	22095D68	3A9944F7	FE78C14F	D164EE63	6F01A173
t=47	3CFA1884	ACBEF4C0	64C7495D	13E12303	C3223A36	3A9944F7	FE78C14F	D164EE63
t=48	3734A72F	3CFA1884	ACBEF4C0	64C7495D	FA9A006C	C3223A36	3A9944F7	FE78C14F
t=49	F8570392	3734A72F	3CFA1884	ACBEF4C0	FA376BB9	FA9A006C	C3223A36	3A9944F7
t=50	BEF7B68B	F8570392	3734A72F	3CFA1884	30B45E08	FA376BB9	FA9A006C	C3223A36
t=51	ECD7F843	BEF7B68B	F8570392	3734A72F	E6203458	30B45E08	FA376BB9	FA9A006C
t=52	FFBFE4A7	ECD7F843	BEF7B68B	F8570392	D3514237	E6203458	30B45E08	FA376BB9
t=53	110C9FBA	FFBFE4A7	ECD7F843	BEF7B68B	054957A0	D3514237	E6203458	30B45E08
t=54	1811A127	110C9FBA	FFBFE4A7	ECD7F843	8D7C1341	054957A0	D3514237	E6203458
t=55	27D632BC	1811A127	110C9FBA	FFBFE4A7	71D850B4	8D7C1341	054957A0	D3514237
t=56	EDF336A5	27D632BC	1811A127	110C9FBA	17C0250D	71D850B4	8D7C1341	054957A0
t=57	5FEF4D7E	EDF336A5	27D632BC	1811A127	409F850C	17C0250D	71D850B4	8D7C1341
t=58	9F1893FC	5FEF4D7E	EDF336A5	27D632BC	05F6290D	409F850C	17C0250D	71D850B4
t=59	BDBF5FCC	9F1893FC	5FEF4D7E	EDF336A5	2B2D4C45	05F6290D	409F850C	17C0250D
t=60	84661E71	BDBF5FCC	9F1893FC	5FEF4D7E	C68224CD	2B2D4C45	05F6290D	409F850C
t=61	4B2A65AE	84661E71	BDBF5FCC	9F1893FC	809E1061	C68224CD	2B2D4C45	05F6290D
t=62	E12108C7	4B2A65AE	84661E71	BDBF5FCC	E08E26DC	809E1061	C68224CD	2B2D4C45
t=63	978AB470	E12108C7	4B2A65AE	84661E71	42F70D1F	E08E26DC	809E1061	C68224CD

8. Menjumlahkan Variabel Kerja Dengan Inisialisasi Nilai *Hash*

Tabel 10. Penjumlahan dengan initial hash value

Variabel	Initial Hash Value	+	Variabel Kerja	Hasil
$H_0^{(0)}$	6A09E667	+	978AB470	01949AD7
$H_1^{(0)}$	BB67AE85	+	E12108C7	9C88B74C
$H_2^{(0)}$	3C6EF372	+	4B2A65AE	87995920
$H_3^{(0)}$	A54FF53A	+	84661E71	29B613AB
$H_4^{(0)}$	510E527F	+	42F70D1F	94055F9E
$H_5^{(0)}$	9B05688C	+	E08E26DC	7B938F68
$H_6^{(0)}$	1F83D9AB	+	809E1061	A021EA0C
$H_7^{(0)}$	5BE0CD19	+	C68224CD	2262F1E6

9. Output

Hasil dari SHA-256 yang saling dirangkaikan dari nilai $H_0^{(0)}$sampai $H_7^{(0)}$ adalah :

01949AD7 || 9C88B74C || 87995920 || 29B613AB || 94055F9E || 7B938F68 || A021EA0C || 2262F1E6

Maka nilai *message digest* yang dihasilkan dari pesan "001011" adalah sebagai berikut:

01949ad79c88b74c8799592029b613ab94055f9e7b938f68a021ea0c2262f1e6

10. Proses Pembangkitan Kunci

a. p dan q adalah bilangan prima, dimana p = 137 dan q = 131.

b. Melakukan perkalian untuk mencari nilai modulus dari kedua bilangan prima yang ditentukan.

$n = p * q$ ..... [3.3]

$= 137 * 131$

$= 17947$

c. Menghitung nilai totien (Ø)

$Ø = (p - 1)(q - 1)$..... [3.4]

$= (137 - 1)(131 - 1)$

$= (136)(130)$

$= 17680$

d. Menentukan nilai *e* dengan syarat memenuhi GCD (*Greater Common Divisor*), dimana (*e*, Ø) = 1, 1 < *e* < Ø. Untuk menentukan nilai *e*, diperlukan persamaan dengan algoritma *Euclidean*.

$r_0 = q_1 r_1 + r_2 , 0 < r_2 < r_1$ ..... [3.5]

$r_1 = q_2 r_2 + r_3 , 0 < r_3 < r_2$

r_n = ...

Maka gcd (e,17680) = 1.

Percobaan pertama nilai e = 2

$r_0 = 2$

$r_1 = 17680$

$r_0 = q_1 r_1 + r_2$

$2 = 0.17680 + 2$

$r_1 = q_2 r_2 + r_3$

$17680 = 8840 . 2 + 0$

Dapat disimpulkan bahwa 2 bukanlah merupakan nilai *e* karena angka pada *r* terakhir sebelum 0 tidak = 1 tetapi 2.

Percobaan ke 2 dengan nilai *e* = 3

$r_0 = q_1 r_1 + r_2$

$3 = 0 . 17680 + 3$

$r_1 = 17680, r_2 = 3$

$r_1 = q_2 r_2 + r_3$

$17680 = 5893 \cdot 3 + 1$

$r_2 = 3, r_2 = 1$

$r_2 = q_3 r_3 + r_4$

$3 = 3 \cdot 1 + 0$

Dari perhitungan diatas dapat dilihat bahwa nilai *r* terakhir sebelum 0 adalah 1, maka nilai *e* adalah 3.

11. Proses Enkripsi Dengan Algoritma RSA

Proses enkripsi dokumen menggunakan kunci publik (*e*,*n*) = 3, 17947. Sebelum melakukan proses enkripsi, konversi *message digest* yang berupa bilangan heksa desimal ke desimal ASCII.

Tabel 11. Konversi Heksadesimal ke Desimal

Heksa Desimal	Desimal	Heksa Desimal	Desimal
0	0	9	9
1	1	4	4
9	9	0	0
4	4	5	5
9	9	5	5
a	10	f	15
d	13	9	9
7	7	e	14
9	9	7	7
c	12	b	11
8	8	9	9
8	8	3	3
b	11	8	8
7	7	f	15
4	4	6	6
c	13	8	8
8	8	a	10
7	7	0	0
9	9	2	2
9	9	1	1
5	5	e	14
9	9	a	10
2	2	0	0
0	0	c	13
2	2	2	2
9	9	2	2
b	11	6	6
6	6	2	2
1	1	f	15
3	3	1	1
a	10	e	14
b	11	6	6

Maka hasil konversi dari *message digest* dari SHA-256 adalah:

0 1 9 4 9 10 13 7 9 12 8 8 11 7 4 13 8 7 9 9 5 9 2 0 2 9 11 6 13 10 11 94 0 5 5 15 9 14 7 11 9 3 8 15 6 8 10 0 2 1 14 10 0 13 2 2 6 2 15 1 14 6

Lakukan enkripsi dengan per karakter heksadesimal.
$M_1 = 0$
$M_2 = 1$
$M_3 = 9$
.
.
$M_{82} = 6$

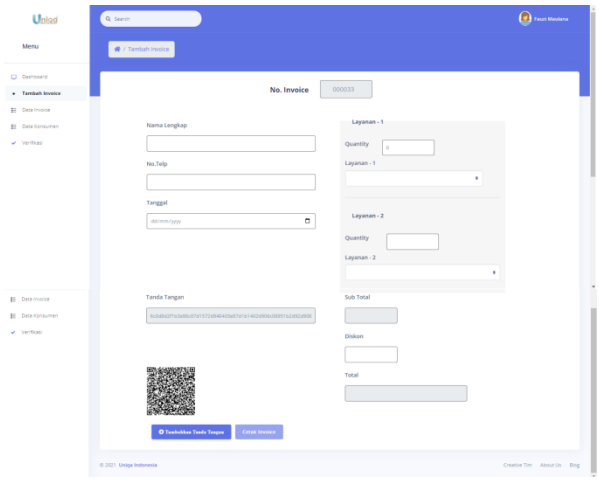
$C_1 = M_1^e \bmod n$ ..... [3.6]
$= 0^3 \bmod 17947$
$= 0$
$C_2 = M_2^e \bmod n$
$= 1^3 \bmod 17947$
$= 1$
$C_3 = M_3^e \bmod n$
$= 9^3 \bmod 17947$
$= 729$
.
.
$C_{82} = M_{82}^e \bmod n$
$= 6^3 \bmod 17947$
$= 216$

Hasil enkripsi dengan algoritma RSA adalah:
0 1 729 64 729 64 729 1000 2197 343 729 1728 512 512 1331 343 64 2197 512 343 729 729 125 729 8 0 8 729 1331 216 1 27 1000 1331 729 64 0 125 125 3375 729 2744 343 1331 729 27 512 3375 216 512 1000 0 8 1 2744 1000 0 2197 8 8 216 8 3375 1 2744 216

Setelah hasil enkripsi diatas di konversi ke heksadesimal, maka *digital signature* yang didapatkan adalah:
012d9402d93e88951572d96c020020053315740895200157 2d92d97d2d98082d9533d811b3e85332d94007d7dd2f2d9ab81575332d91b200d2fd82003e8081ab83e8089588d88d2f1ab8d8

3.4 Pengujian

Proses pengujian dilakukan dengan memasukkan data-data yang wajib diisi agar sejalan dengan pembentukan digital signature. No.invoice otomatis berurutan dilakukan sistem, bersamaan dengan proses untuk mendapatkan *digital signature* dan mengubahnya ke QR Code.



Gambar 2. Pembuatan e-Invoice

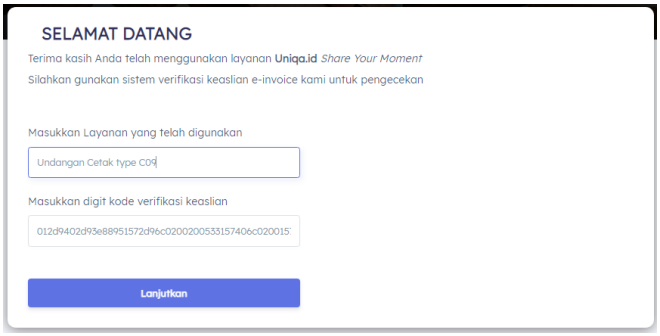
Setelah dilakukannya penambahan data e-*invoice*, data tersimpan ke database. Adapun *e-invoice* yang hasilkan seperti gambar berikut:

Gambar 3. E-Invoice

Untuk dapat membaca QR Code untuk menghasilkan teks yang disembunyikan diperlukan pihak ketiga sebagai reader QR Code yaitu Google Lens. Dengan melakukan scan pada QR Code, Google Lens akan membaca dan mengeluarkan teks hasil dari pembacaan QR Code.

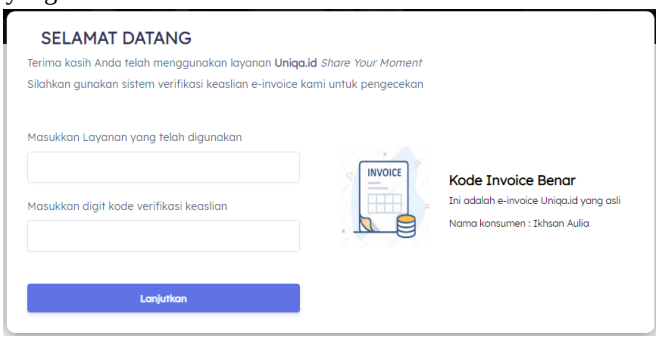
Gambar 4. Proses Scan QR Code Dengan Google Lens

Pengecekan dengan memasukkan digit kode verifikasi dari hasil scan Google Lens ke *Form* Verifikasi Keaslian yang bisa digunakan untuk pengguna jasa Uniqa Digital Invitation.



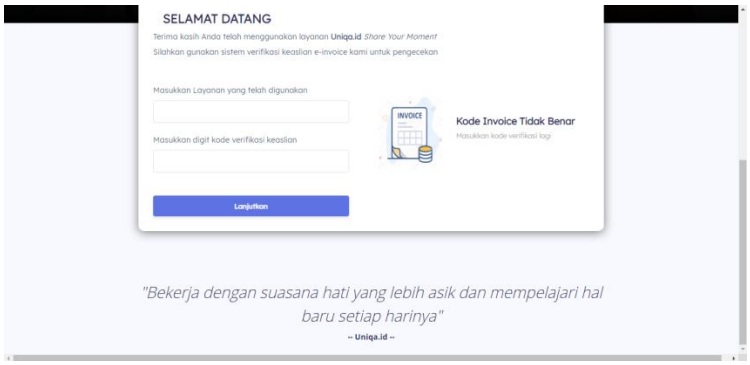
Gambar 5. Verifikasi Keaslian

Sistem akan mengeluarkan informasi valid dan menampilkan nama konsumen jika kode verifikasi yang dimasukkan sesuai dengan yang terdaftar.



Gambar 6. Kode Verifikasi Benar

Jika kode verifikasi yang dimasukkan tidak sesuai atau tidak valid, sistem akan memunculkan informasi seperti gambar berikut:



Gambar 7. Kode Verifikasi Tidak Valid

Berdasarkan hasil pengujian yang dilakukan, *e-invoice* yang diberikan tanda tangan digital dapat memberikan bukti autentikasi terkait digit kode verifikasi yang sudah melalui proses fungsi hash hingga dapat terdeteksi oleh sistem. Tanda tangan digital yang terdapat pada *e-invoice* memiliki nilai kode verifikasi yang berbeda-beda. Ini membuat *e-invoice* menjadi lebih mudah diketahui jika *e-invoice* yang dimiliki sudah terjadi modifikasi hingga fabrikasi.

4. KESIMPULAN

Adapun kesimpulan dari penelitian ini yaitu sebagai berikut :

1. Berdasarkan pengujian dan implementasi sistem digitak signature dapat melakukan proses membuat tanda tangan digital hingga memapu digunakan untuk menjaga keaslian data *e-invoice* dengan menerapkan algoritma SHA-256 dan RSA.
2. Berdasarkan hasil analisa, algoritma SHA-256 dan RSA dapat diterapkan dalam menjaga keaslian data *e-invoice* dengan hasil *digital siganture* yang menghasilkan kode acak yang rumit.
3. Berdasarkan hasil pengujian, efektifitas dari sistem *digital signature* yang dirancang terhadap masalah yang diteliti sudah sangat baik.

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] F. Nuraeni, Y. H. Agustin, D. Kurniadi, and I. D. Ariyanti, "Implementasi Skema QR-Code dan Digital Signature menggunakan Kombinasi Algoritma RSA dan AES untuk Pengamanan Data Sertifikat Elektronik," *SNTIKI*, pp. 43–52, 2020.

[2] Dedy Alamsyah, "Pengembangan Purwarupa Sistem Proteksi Hybrid Keaslian Faktur Elektronik (E-Invoice) Pada E-Bisnis Menggunakan QR Code, Steganografi Dan Kriptografi," *J. Tek.*, vol. 5, No 2, 2016.

[3] P. Saha, "A Comprehensive Study On Digital Signature For Internet Security," *Accent. Trans. Inf. Secur.*, vol. 1, no. 1, pp. 1–6, 2016.

[4] Y. Anshori, A. Y. Erwin Dodu, and D. M. P. Wedananta, "Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) Pada Tanda Tangan Digital," *Techno.Com*, vol. 18, no. 2, pp. 110–121, doi: 10.33633/tc.v18i2.2166.

[5] F. Nuraeni, Y. H. Agustin, and I. M. Muharam, "Implementasi Tanda Tangan Digital Menggunakan RSA dan SHA-512 Pada Proses Legalisasi Ijazah," *Knsi*, pp. 864–869, 2018.

[6] W. W. Widiyanto, "Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (Rad)," *J. Inf. Politek. Indonusa Surakarta ISSN*, vol. 4, no. 1, pp. 34–40, 2018, [Online]. Available: http://www.informa.poltekindonusa.ac.id/index.php/informa/article/view/34.

[7] R. Simarmata, Janner;, Sriadhi; Rahim, "KRIPTOGRAFI," in *KRIPTOGRAFI; Teknik Keamanan Data dan Informasi*, 1st ed., M. Kiki, Ed. Yogyakarta: CV. ANDI OFFSET, 2019.

[8] Z. Panjaitan, E. F. Ginting, and Y. Yusnidah, "Modifikasi SHA-256 dengan Algoritma Hill Cipher untuk Pengamanan Fungsi Hash dari Upaya Decode Hash," *J. SAINTIKOM (Jurnal Sains Manaj. Inform. dan Komputer)*, vol. 19, no. 1, pp. 53–61, 2020.

[9] S. Sulastri, "Implementasi Enkripsi Data Message Digest Algorithm 5 (MD5) Dan Secure Hash Algorithm (SHA-256) Pada Sistem Penjadwalan Karyawan Agrowisata Setya Aji Flower Farm Bandungan," 2019.

[10] M. Y. Simargolang, "Implementasi Kriptografi RSA Dengan PHP," *Teknol. Inf.*, vol. 1, no. 1, pp. 1–10.

[11] I. A. Egi Cahyo Prabowo, "Penerapan Digital Signature Dan Kriptografi Pada Otentikasi Sertifikat Tanah Digital," *J. Ilm. Komput. dan Inform.*, vol. 6, no. 2.

BIBLIOGRAFI PENULIS

	**Fauzi Maulana Rangkuti** Lahir pada tahun 1998 di Medan, Sumatera Utara. Saat ini sedang menempuh studi Sistem Informasi di STMIK Triguna Dharma. Sejak tahun 2016 hingga saat ini bekerja sebagai Wiraswasta di Medan. Aktif sebagai relawan kemanusiaan. Memiliki keahlian dalam membangun Aplikasi Mobile Android dan Website. Menyelesaikan Program Kreativitas Mahasiswa (PKM-P) bersama rekan lainya pada tahun 2020.
	**Nurcahyo Budi Nugroho** Dosen tetap STMIK Triguna Dharma bidang Sistem Informasi, beliau aktif mengampu mata kuliah bidang pemrograman Website, Dekstop dan Mobile. Aktif dalam pengembangan mutu mahasiswa dalam Bidang Pemrograman dan Teknik Algoritma Pemrograman.
	**Zaimah Panjaitan** Lahir pada tahun 1989 di Sei Paham. Saat ini merupakan Dosen tetap STMIK Triguna Dharma. Bidang riset yang ditekuni saat ini adalah Keamanan Komputer dan Artificial Intelligence. Beliau pernah bekerja di PenPes (PP) Baitussalam Siantar dan International Boarding School Ar-Raudhatul Hasanah sebagai staff penelitian dan pengembangan atau IT dan Jurnalistik dan seorang pengajar pada tahun 2010-2015. Tamat 2015 Strata 1 Sistem Informasi STMIK Triguna Dharma dan 2018 Strata 2 Magister Ilmu Komputer Universitas Putra Indonesia YPTK Padang.