Jurnal Matematika Integratif. p-ISSN:1412-6184, e-ISSN:2549-903
Vol.18, No.1(2022), pp. 27–39. doi:10.24198/jmi.v18.n1.38343.27-39

Verifikasi Tanda Tangan Elektronik dengan Teknik Otentikasi Berbasis Kriptografi Kunci Publik Sistem Menggunakan Algoritma Kriptografi Rivest-Shamir-Adleman

Melina[1*], Sukono[2], Herlina Napitupulu [2], Valentina Adimurti Kusumaningtyas [3]

[1]Program Studi S-1 Informatika, Fakultas Sains dan Informatika Universitas Jenderal Achmad Yani, melina@lecture.unjani.ac.id
[2]Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran, sukono@unpad.ac.id, herlina@unpad.ac.id,
[3]Jurusan Kimia, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani, valentina.adimurti@lecture.unjani.ac.id

[*]Corresponding Author

Abstrak

Dalam upaya menekan penyebaran COVID-19, hampir seluruh negara melakukan lockdown. Sebagian besar aktivitas menjadi bersifat *work from home* (WFH) sehingga memicu peningkatan transaksi *online*. Transaksi *online* yang melibatkan informasi penting memerlukan tanda-tangan elektronik (TTE), yang merupakan alat autentikasi dan verifikasi. Proses keabsahan pengirim, keaslian, dan anti penyangkalan dengan cepat, dan akurat menjadi suatu keharusan. Prosedur yang digunakan untuk membuktikan keaslian TTE, keaslian identitas penandatangan, dan anti-penyangkalan dapat dilakukan dengan teknik otentikasi. Tujuan penelitian ini adalah mengusulkan verifikasi TTE dengan teknik otentikasi berbasis kriptografi kunci-publik sistem dengan cara membalikkan peran kunci privat dan kunci publik pada algoritma kriptografi Rivest-Shamir-Adleman. Informasi elektronik yang melekat pada TTE akan disimpan kedalam pesan M, dengan menggunakan fungsi hash terhadap M menghasilkan $h = H(M)$. Nilai h akan di enskripsi untuk menjadi TTE dengan kunci privat PR (d, n). Tanda tangan diverifikasi untuk membuktikan keotentikannya dengan mendekripsi TTE menggunakan kunci publik PU $(e,\ n)$, sehingga mendapatkan nilai h. Nilai h dibandingkan dengan nilai h' yang diperoleh dari nilai fungsi hash pesan M dari TTE yang diperiksa. Jika $h = h'$ berarti TTE otentik. Jika ukuran digit $d > e$ maka waktu penandatangan akan lebih lama dari pada waktu yang diperlukan untuk verifikasi, begitu juga sebaliknya.

Kata kunci: COVID-19, hash, kriptografi kunci-publik sistem, otentikasi.

2000 Mathematics Subject Classification: 68M25
Received: 12-02-2022, Revised: 09-04-2022, Accepted: 17-04-2022.

Abstract

To suppress the spread of COVID-19, almost all countries have been locked down. Most activities become work from home (WFH) so that triggers an increase in online transactions. Online transactions involving critical information require an electronic signature (DS), which is an authentication and verification tool. A fast and accurate process of sender legitimacy, authenticity, and anti-disclaimer is a must. The procedures used to prove the authenticity of DS, authenticity of the signer's identity, and anti-repudiation can be carried out with authentication techniques. The purpose of this study is to propose a DS verification with an authentication technique based on system public-key cryptosystems by reversing the role of the private key and public key in the Rivest-Shamir-Adleman cryptographic algorithm. The electronic information attached to the DS will be stored in the message M, using a hash function against M to produce $h = H(M)$. The value of h will be encrypted to be a DS with the private key PR $(d,\ n)$. The signature is verified to prove its authenticity by decrypting the DS using the public key PU $(e,\ n)$, thus obtaining the value of h. The value of h is compared with the value of h′ which is obtained from the value of the message hash function M of the examined DS. If $h = h'$ means the DS is authentic. If the digit size is $d > e$, the signing time is longer than the time required for verification, and vice versa.

Keywords: Authentication, COVID-19, hash, public-key cryptosystems.

1. Pendahuluan

Pandemi Corona virus 2019 (COVID-19) telah menyerang setiap negara [1, 2, 3]. Pandemi ini menyebabkan perubahan dalam kehidupan manusia yang disebut *new normal* [4]. Segala upaya dilakukan untuk menekan penyebaran COVID-19. Salah satunya adalah melakukan *lockdown* yang dilakukan hampir seluruh negara. *Lockdown* di Indonesia dikenal juga dengan istilah pembatasan sosial berskala besar (PSBB), dan juga Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) [5, 6]. Salah satu program dari *lockdown* adalah melakukan kegiatan bersifat *work from home* (WFH) [7]. WFH memicu peningkatan transaksi *online*. Selama pandemi penjualan *e-commerce* meningkat 26% dengan konsumen baru sebesar 51% [8, 9], dimana terjadi peningkatan jumlah pengguna internet sekitar 25,5 juta atau 8,9% jika dibandingkan pada tahun 2019 [10]. Akibat WFH, semuanya dilakukan secara *online*, termasuk juga sektor pendidikan, dimana proses belajar-mengajar dilakukan secara *online*, menyebabkan kebutuhan tanda tangan elektronik (TTE) semakin mendesak.

Tanda tangan dalam bahasa Inggris adalah *signature* bersumber dari bahasa Latin yaitu: *signare* yang artinya adalah tanda, atau disebut juga dengan paraf. Paraf juga dikatakan tulisan tangan, yang diberi variasi tulisan tertentu dari nama dan inisial seseorang atau tanda identifikasi lainnya yang ditulis pada dokumen yang juga merupakan bukti dari identitas dan keinginan. TTE adalah sebuah kombinasi unik dari fungsi hash dan enkripsi dengan menggunakan metode asimetris [11]. Di Indonesia TTE juga diatur pada Peraturan Pemerintah No.82 Tahun 2012 tentang penyelenggaraan sistem transaksi elektronik. Pada PP tersebut disebutkan bahwa TTE berfungsi sebagai alat autentikasi dan verifikasi atas identitas penandatangan, serta keutuhan dan keautentikan informasi elektronik (IE) [12, 13]. Selain itu setiap dokumen elektronik yang disebarkan atau didistribusikan melalui internet atau media elektronik persetujuan yang diakui adalah berupa TTE tersebut, bukan tanda tangan basah hasil *scan* dari *scanner* [14]. Berdasarkan pada pasal 1 ayat (12) undang undang no 11 tahun 2008,

TTE merupakan tanda tangan yang terdiri atas IE yang dilekatkan, terasosiasi atau terkait dengan IE lainnya yang akan digunakan sebagai alat verifikasi dan autentikasi [15].

Algoritma kriptografi Rivest-Shamir-Adleman (RSA) dijabarkan pada tahun 1977 oleh Ron Rivest, Adi Shamir dan Len Adleman [16]. RSA itu sendiri merupakan singkatan nama Rivest, Shamir, dan Adleman. RSA merupakan algoritma kriptografi yang menggunakan kunci asymmetric [17]. Sistem kriptografi asimetris biasanya juga dikenal dengan kriptografi kunci publik sistem (KKPS) [18]. Pada algoritma kriptografi RSA ada sepasang kunci yaitu kunci publik serta kunci privat. Pada proses enkripsi, dekripsi RSA berdasarkan konsep bilangan-bilangan prima dan juga aritmetika modulo. Kedua kunci enkripsi dan dekripsi merupakan bilangan bulat positif. Kunci publik bersifat tidak dirahasiakan dan diberikan kepada publik umum, sebaliknya kunci privat bersifat rahasia dan pribadi. Kunci privat dibangkitkan dari dua bilangan prima yaitu p dan q, bersama-sama dengan kunci enkripsi [19]. Sulitnya memfaktorkan bilangan yang besar menjadi faktor faktor prima merupakan keamanan algoritma RSA yang andal. Selama konsep pemfaktoran bilangan besar menjadi faktor-faktor prima belum terpecahkan, maka selama itu keamanan algoritma RSA tetap andal dan terjamin [20].

Fungsi hash merupakan sebuah fungsi yang bersifat satu arah dan menghasilkan fungsi unik yang berjumlah tetap untuk setiap data yang dimasukan pada fungsi hash tersebut [21]. Fungsi hash merupakan sebuah fungsi yang menerima masukkan berupa *string* lalu diproses sesuai dengan standar fungsi hash tersebut yang kemudian akan menghasilkan string dengan panjang yang tetap. Luaran yang dihasilkan fungsi hash tidak akan bisa dikembalikan lagi menjadi data aslinya [22]. Pesan yang berukuran sembarang diubah oleh fungsi hash menjadi *massage digest* (MD) atau pesan singkat yang berukuran tetap. MD disebut juga nilai hash (*hash value*) dari fungsi hash, Fungsi hash kriptografi digunakan untuk TTE. Kombinasi fungsi hash dengan algoritma KKPS digunakan untuk menghasilkan TTE. TTE diperoleh dari hasil enkripsi terhadap MD yang dihasilkan fungsi hash dengan cara membalikkan peran kunci privat dan kunci publik pada algoritma kriptografi RSA, sehingga kerahasiaan pesan dan otentikasi dapat dicapai sekaligus. Enkripsi pada TTE menggunakan kunci privat, sedangkan proses deskripsi menggunakan kunci publik. TTE dikirim bersama IE yang tidak dienkripsi sehingga dapat digunakan untuk proses verifikasi. Seiring kemajuan dan perkembangan yang sangat pesat pada teknologi sistem informasi, serta meningkatnya kemampuan *hypertext markup language* (HTML) dalam bentuk basis-data dan grafik [23], maka verifikasi TTE dengan teknik otentikasi berbasis KKPS menggunakan algoritma RSA dapat dilakukan secara *online*. Algoritma kriptografi RSA dapat digunakan untuk otentikasi TTE, sehingga penyangkalan terhadap sesuatu aksi dapat dicegah, dan mempunyai tingkat keamanan berlapis karena memakai kunci privat dan publik dalam proses enkripsi dan deskripsinya [24]. Pada verifikasi otentikasi data, pengirim dan penerima membandingkan kode hash dan memeriksa apakah itu asli. Pesan itu otentik ketika pesan yang diambil oleh penerima sama dengan pesan awalnya ditandatangani [25]. Pada penelitian ini kami membangkitkan MD mengunakan algoritma hash *message digest* 5 (MD5), yang merupakan fungsi hash bawaan bahasa pemograman *hypertext prepocessor* (PHP) versi 7 [26].

Penelitian terkait dengan TTE menggunakan algoritma RSA telah banyak dilakukan oleh peneliti terdahulu seperti penelitian yang dilakukan oleh Rezania (2016) yang mengimplementasikan algoritme Keccak dan RSA pada TTE dan membandingkan dengan MD5. Hasil pengujian dari implementasi kedua algoritma menunjukkan TTE memerlukan waktu yang relatif singkat serta dapat menjamin keamanan pada aspek integritas, autentikasi, dan *non-repudiation* [17]. Penelitian oleh Rizky (2017) mengkombinasikan tiga algoritma untuk TTE yaitu RSA, *vigenere cipher* dan MD5 dan diuji dengan berbagai serangan untuk mengukur keandalan TTE seperti *blurring, salt, pepper*, dan *gaussian filter*. Berdasarkan hasil *attack*, perubahan terkecil yang terjadi pada *blurring attack* memiliki *Peak Signal to Noise Ratio* (PSNR) yang sangat baik yaitu 86.7532 dB. Hasil penelitian membuktikan bahwa sedikit perubahan pada citra dan nama *file* dapat mempengaruhi hasil validasi. Sehingga metode yang diusulkan cocok untuk otentikasi citra [27]. Penelitian oleh Jarah Jihan dkk (2018), membandingkan waktu komputasi RSA dan *digital signature algorithm* (DSA) dengan beberapa bit dan

memilih bit mana yang lebih baik digunakan dan menggabungkan algoritma RSA dan DSA untuk meningkatkan keamanan data. Hasil penelitian menunjukkan bahwa dari hasil simulasi, RSA 1024 digunakan untuk proses enkripsi dan ditambahkan TTE menggunakan DSA 512, sehingga pesan yang dikirim tidak hanya dienkripsi tetapi juga memiliki TTE untuk proses otentikasi data [28]. Penelitian oleh Kritsanapong (2020) menggunakan TTE RSA untuk menandatangani e-sertifikat agar tidak ada pemalsuan dengan dua aplikasi mengelola *e-certificate*. Aplikasi pertama adalah aplikasi penandatanganan untuk menandatangani sub gambar yang hanya mencantumkan nama peserta dalam e-sertifikat, Pada umumnya file TTE dipisahkan dari e-sertifikat. Hasil penelitian menunjukkan akurasi 100%, dan proses penandatanganan dan pemeriksaan selesai dengan cepat, terutama ketika aplikasi penandatanganan diterapkan dengan *Chinese Remainder Theorem* (CRT) sehingga metode yang diusulkan adalah salah satu solusi terbaik untuk melindungi e-sertifikat dari pemalsuan oleh penyusup [29]. Penelitian oleh I made Ari Dwi Suta Atmaja dkk (2020), menggunakan kriptografi dengan algoritma RSA untuk mengamankan dokumen dengan dienkripsi terlebih dahulu sebelum dikirim dengan email. Pada proses enkripsi akan dihasilkan kunci publik dan kunci privat yang dapat dikirim secara terpisah dengan mengirimkan dokumen *digital* terenkripsi. Proses dekripsi untuk dokumen *digital* dilakukan dari bagian penerima dokumen menggunakan kunci pribadi yang dihasilkan dalam proses enkripsi. Hasil penelitian menunjukkan semakin lama dan semakin besar ukuran input, semakin lama waktu yang dibutuhkan untuk enkripsi [30]. Penelitian oleh Amer Sharif, dkk (2021) yang mengkombinasikan RSA dengan *digital signature* (DS) berbasis web dengan fungsi *hash, secure hash algorithm* 3 (SHA-3) untuk mengamankan integritas file PDF menggunakan bahasa pemrograman PHP. Hasil penelitian adalah sistem berbasis web yang dapat menjamin keaslian, tidak ada penolakan dan integritas file PDF [31].

Berdasarkan penelitian terdahulu, terdapat gap yang menunjukkan bahwa masih kurangnya penelitian yang mengkaji tentang verifikasi TTE dengan teknik otentikasi berbasis KKPS menggunakan algoritma kriptografi RSA. Oleh karena itu, penulis tertarik melakukan penelitian verifikasi TTE dengan teknik otentikasi berbasis KKPS dengan algoritma kriptografi RSA. IE yang melekat pada TTE akan disimpan kedalam pesan M, dengan menggunakan fungsi hash terhadap M menghasilkan MD atau nilai hash (h). Nilai h akan di enskripsi untuk menjadi TTE dengan kunci privat. Tanda tangan diverifikasi untuk dibuktikan keotentikannya dengan mendekripsi TTE menggunakan kunci publik, sehingga mendapatkan nilai h. Nilai h akan dibandingkan dengan nilai h' yang diperoleh dari nilai fungsi hash pesan M dari TTE yang diperiksa. Jika $h = h'$ berarti TTE otentik, maka TTE adalah asli, original, dan penandatangan tidak dapat menyangkal bahwa itu tandatangannya, dan proses otentikasi TTE dapat dilakukan oleh semua pihak.

2. Metode Penelitian

Berdasarkan uraian latar belakang di atas, maka dapat diambil suatu rumusan masalah yaitu, bagaimana verifikasi TTE dengan teknik otentikasi berbasis KKPS dengan menggunakan algoritma kriptografi RSA, sehingga IE yang melekat pada TTE dapat dilindungi sekaligus dapat autentikasi dengan cepat, dan diidentifikasi keabsahannya. Algoritma RSA terdiri dari 3 tahap [21], yaitu: algoritma pembangkitan kunci, enkripsi, dan deskripsi. Dasar algoritma RSA adalah teorema Euler [32, 33], sebagai berikut:

$$a^{\emptyset(n)} \equiv 1 \ mod \ (n) \tag{1}$$

dimana,

(1) $\emptyset$ relatif terhadap n, dan

(2) $\emptyset\,(n) = n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\left(1-\frac{1}{p_3}\right)\dots\left(1-\frac{1}{p_r}\right)$
$p_1,\ p_2,\ p_3, \dots,\ p_r$ adalah faktor prima dari n.

Berdasarkan sifat dari $a^k \equiv b^k \ mod \ (n)$ dengan $k \geq 1$, maka persamaan (2) jadi

$$a^{k\emptyset(n)} \equiv 1^k \ \ (mod \ n)\,. \tag{2}$$

Jika a diganti menjadi m, maka

$$m^{k\emptyset(n)} \equiv 1 \ (mod\ n)\,. \tag{3}$$

Berdasarkan sifat $ac \equiv bc\ (mod\ n)$, bila persamaan (3) dikalikan dengan m, maka:

$$m^{k\emptyset(n)+1} \equiv m\ (mod\ n) \tag{4}$$

dimana m relatif prima terhadap n, karena

$$e.d \equiv 1\ (mod\ \emptyset\,(n)) \tag{5}$$

atau

$$e.d \equiv k\emptyset\,(n) + 1. \tag{6}$$

Subsitusikan persamaan (6) kedalam persamaan (4), maka

$$m^{e.d} \equiv m\ (mod\ n) \tag{7}$$

atau

$$(m^e)^d \equiv m \ \ (mod\ n)\,. \tag{8}$$

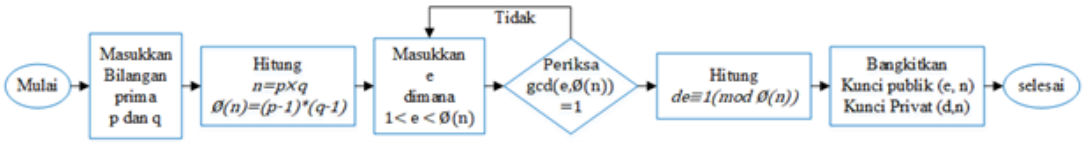
Artinya jika m pangkat e dipangkatkan lagi dengan d maka menghasilkan m kembali. Berdasarkan persamaan (8) dapat dirumuskan enkripsi sebagi berikut [19]:

$$c = m^e\,(mod\ n)\,, \tag{9}$$

dan dekripsi dirumuskan sebagai berikut:

$$m = c^d\,(mod\ n)\,. \tag{10}$$

Kunci enkripsi maupun dekripsi merupakan pasangan kunci yang paling penting dalam algoritma kriptografi RSA. Berikut Gambar 1, memperlihatkan tahapan membangkitkan sepasang kunci publik dan kunci privat dengan metode kriptografi RSA sebagai berikut:



Gambar 1. Tahapan membangkitkan kunci dengan algoritma kriptografi RSA

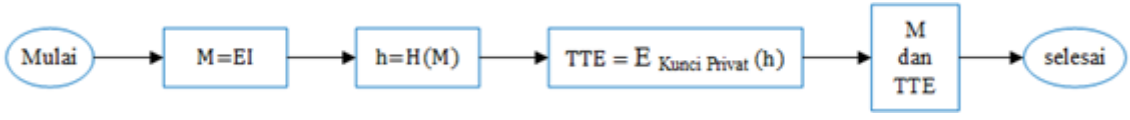
Mekanisme TTE dibagi dua proses, yaitu penandatanganan dan verifikasi TTE. Proses penandatangan dimulai dengan cara menghitung MD (h) dari pesan M, menggunakan fungsi hash satu arah H, dirumuskan sebagai berikut:

$$h = H(M). \tag{11}$$

TTE didapat dari enkripsi MD h dengan menggunakan kunci privat. Proses penandatangan ini, dirumuskan sebagai berikut:

$$TTE = E_{\text{Kunci Privat}}\,(h)\,. \tag{12}$$

Tahapan proses penandatangan TTE ini dapat dilihat pada Gambar 2, seperti berikut.



Gambar 2. Proses penandatangan TTE

Hasil dari proses penandatangan ini adalah sebuah pesan M yang berisi struktur dan data-data IE, kemudian dilekatkan atau dikirim bersama dengan TTE sehingga menandakan bahwa pesan M telah ditandatangani dengan TTE. Kemudian TTE dapat di verifikasi untuk

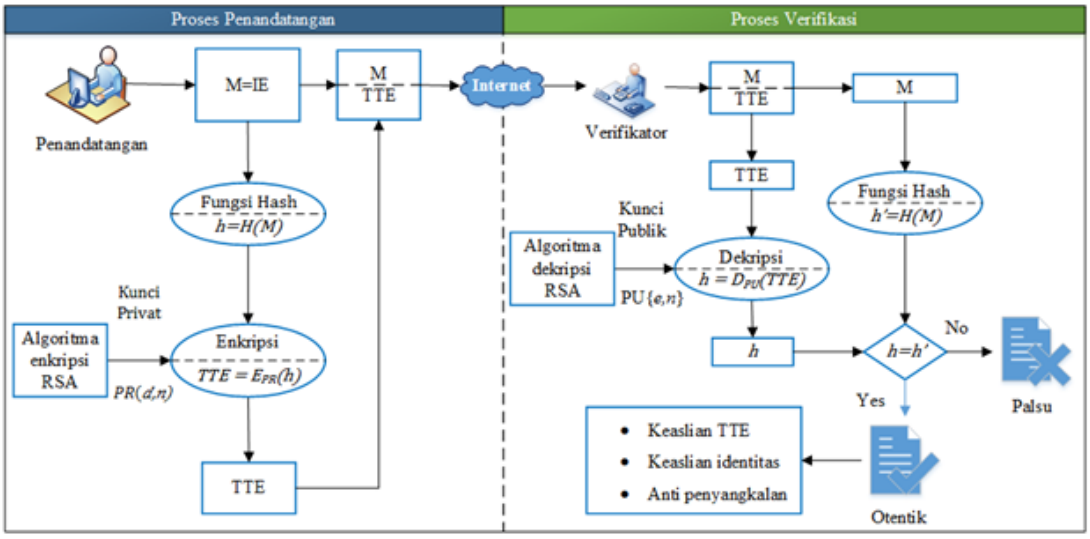
membuktikan ke otentikasi dengan mendeskripsi dengan kunci publik, dan menghasilkan h, proses penandatangan ini dirumuskan sebagai berikut:

$$h = D_{Kunci\ Publik}\,(TTE)\,. \tag{13}$$

Disisi lain, dengan menggunakan fungsi hash, maka dihitung MD dari pesan M. Pesan M ini didapat dari dari TTE yang akan diperiksa. Pesan M berisi field dari IE. Hasil dari proses ini adalah nilai h', yang dirumuskan sebagai berikut:

$$h' = H(M) \tag{14}$$

Jika $h = h'$, berarti pesan yang diterima adalah pesan asli dan keaslian identitas pengirim. Penandatangan tersebut yang membuat kunci privat dan kunci publik, sehingga penandatangan tidak dapat menyangkal telah menandatangan pesan tersebut. Metode yang dapat digunakan pada pembuktian: keaslian TTE, keaslian identitas pengirim, dan penandatangan tidak dapat menyangkal isi IE dari TTE disebut juga teknik otentikasi [32]. Ilustrasi skema teknik otentikasi TTE berbasis KKPS dengan menggunakan algoritma kriptografi RSA ini dapat dilihat pada Gambar 3, sebagai berikut.



Gambar 3. Skema teknik otentikasi TTE berbasis KKPS

3. Hasil dan Pembahasan

Tahap pertama dalam proses TTE adalah mengisi EI yang nantinya dilampirkan pada tanda tangan. EI merupakan struktur yang berisi informasi yang menjelaskan perihal tanda tangan, yang merupakan kumpulan data seperti: nomor tanda tangan, tanggal penandatangan, judul tanda tangan, dan nama penandatangan. Ilustrasi dari struktur IE dapat dilihat pada Gambar 4, sebagai berikut.

Informasi Elektronik	
No	: TTE/001
Tanggal	: 10-Januari-2022
Judul	: Daftar Nilai UAS Matematika
Oleh	: Alice

Gambar 4. Ilustrasi Struktur IE

Strukur dari IE tersebut kemudian disimpan dalam variabel IE$, seperti berikut

$$IE\$ = "TTE/001; 10 - Januari - 2022; AbsenUASMatematika; Alice"$$

Dengan menggunakan persamaan (11), IE terlebih dahulu diubah menjadi bentuk yang ringkas yang disebut MD. Proses menghitung nilai hash ini lakukan dengan fungsi bawaan bahasa pemograman PHP, maka didapat:

$$h = c67f50517185d171b8098fed309f982c$$

Selanjutnya, dengan menggunakan persamaan (12) nilai h dienkripsi dengan algoritma kriptografi RSA, yang berfungsi untuk menjadi MD sebagai TTE. Proses enkripsi dan deskripsi pada algoritma RSA membutuhkan sepasang kunci privat dan rahasia.

3.1. **Algoritma Pembangkit Kunci.** Misalkan Alice membangkitkan kunci privat dan kunci publik dengan memilih dua bilangan prima yaitu: $P = 71$ dan $q = 79$. Selanjutnya hitung nilai n.

$$n = p \times q = 5609$$

maka,

$$\emptyset(n) = (p-1)(q-1) = 5460$$

Pilih bilangan bulat e, dimana e relatif prima dengan $\emptyset(n)$, jadi $\gcd(\emptyset(n), e) = 1$, dimana $1 < e < \emptyset(n)$, kami pilih $e = 19$, karena 19 relatif prima dengan 5460. Bukti $\gcd(5460, 19) = 1$, dapat dilihat pada Tabel 1, sebagai berikut.

Tabel 1. Penentuan $\gcd(5460, 19) = 1$

No	Proses	Hasil
1	$5460 \ mod \ 19$	7
2	$19 \ mod \ 7$	5
3	$7 \ mod \ 5$	2
4	$5 \ mod \ 2$	1
5	$2 \ mod \ 1$	0

Pada langkah ke 5, proses berhenti karena $2 \ mod \ 1 \ = 0$, maka didapat $\gcd(5460, 19) = 1$. Berdasarkan persamaan (6), yang ekivalen dengan $e.d = 1 + k\emptyset(n)$ sehingga d dapat dihitung rumus:

$$d = \frac{1 + k\emptyset(n)}{e}$$
$$d = \frac{1 + 8 \times 5460}{19} = 2299.$$

Hasil dari algoritma pembangkit kunci maka didapat sepasang kunci, yaitu:

(1) Kunci privat, PR ($d = 2299, \ n = 5609$),
(2) Kunci publik, PU($e = 19, \ n = 5609$).

3.2. **Proses Penandatangan.** TTE ditandatangani oleh Alice dengan cara mengenkripsi nilai h yang telah didapat sebelumya, yaitu:

$$h = c67f50517185d171b8098fed309f982c.$$

Proses penandatangan dirumuskan berdasarkan persamaan (12). Nilai h dienkripsi menggunakan kunci privat, PR($d = 2299, \ n = 5609$). Jika nilai h dikonversi kedalam sistem desimal pengkodean *American Standard Code for Information Interchange* (ASCII), hasilnya dapat juga dilihat pada Tabel 2.

Nilai h yang telah dikonversi ke bentuk desimal akan di pecah menjadi blok-blok plainteks, misalnya m dipecah menjadi 32 blok berukuran 3 *digit*, dengan blok $m_1, \ m_2, \ m_3, \ m_4 \ldots, \ m_n$ sedemikian rupa sehingga setiap blok-blok mempresentasikan nilai di dalam selang $[0, \ n-1]$.

Tabel 2. Konversi nilai h ke sistem desimal pengkodean ASCII

No	Nilai	Desimal	No	Nilai	Desimal	No	Nilai	Desimal	No	Nilai	Desimal
1	c	099	9	7	055	17	b	098	25	3	051
2	6	054	10	1	049	18	8	056	26	0	048
3	7	055	11	8	056	19	0	048	27	9	057
4	f	102	12	5	053	20	9	057	28	f	102
5	5	053	13	d	100	21	8	056	29	9	057
6	0	048	14	1	049	22	f	102	30	8	056
7	5	053	15	7	055	23	e	101	31	2	050
8	1	049	16	1	049	24	d	100	32	c	099

Pada Tabel 3, akan diperlihatkan pemecahan blok-blok plainteks menjadi 32 blok berukuran 3 digit, sebagai berikut.

Tabel 3. Pemecahan plainteks menjadi 32 blok berukuran 3 *digit*

No	Blok	No	Blok	No	Blok	No	Blok
1	$m_1 = 099$	9	$m_9 = 055$	17	$m_{17} = 098$	25	$m_{25} = 051$
2	$m_2 = 054$	10	$m_{10} = 049$	18	$m_{18} = 056$	26	$m_{26} = 048$
3	$m_3 = 055$	11	$m_{11} = 056$	19	$m_{19} = 048$	27	$m_{27} = 057$
4	$m_4 = 102$	12	$m_{12} = 053$	20	$m_{20} = 057$	28	$m_{28} = 102$
5	$m_5 = 053$	13	$m_{13} = 100$	21	$m_{21} = 056$	29	$m_{29} = 057$
6	$m_6 = 048$	14	$m_{14} = 049$	22	$m_{22} = 102$	30	$m_{30} = 056$
7	$m_7 = 053$	15	$m_{15} = 055$	23	$m_{23} = 101$	31	$m_{31} = 050$
8	$m_8 = 049$	16	$m_{16} = 049$	24	$m_{24} = 100$	32	$m_{32} = 099$

Selanjutnya plainteks akan dienkripsi dengan menggunakan algoritma kriptografi RSA dengan cara membalikkan peran kunci privat dan kunci publik pada algoritma kriptografi RSA. Algoritma enkripsi kriptografi RSA menggunakan persamaan (9). Proses enkripsi berbasis KKPS menggunakan kunci privat PR($d = 2299,\ n = 5609$), sehingga persamaan (9) akan dirubah menjadi persamaan:

$$c_i = m_i^d \,(mod\ n)\,. \tag{15}$$

Dengan menggunakan persamaan (15) plainteks akan dienkripsi. Plainteks akan dibagi menjadi blok-blok $m_1,\ m_2,\ m_3,\ \ m_4 \ldots,\ m_n$. Setiap blok-blok mempresentasikan nilai di dalam selang $[0, 5609 - 1]$. Hasil enkripsi dapat dilihat pada Tabel 4.

Luaran dari proses enkripsi adalah chiperteks yang merupakan gabungan masing-masing blok-blok c_i. Maka diperoleh

chiperteks = 1201 4853 3531 1050 4494 5299 4494 5581 3531 5581 4480 4494 1884 5581 3531 5581 151 4480 5299 3626 4480 1050 2931 1884 1514 5299 3626 1050 3626 4480 505 1201.

Hasil enkripsi yang berupa chiperteks ini disebut juga TTE. Sehingga dapat dikatakan bahwa IE tersebut telah ditandatangani dalam bentuk algoritma enkripsi oleh penandatangan.

3.3. **Verifikasi.** Tanda tangan akan diverifikasi untuk membuktikan keotentikannya dengan menggunakan teknik otentikasi. Pada Gambar 5, akan diperlihatkan ilustrasi TTE beserta struktur IE yang melekat pada TTE tersebut.

Dari Gambar 5, dapat ditentukan IE yang melekat pada TTE adalah:

$$IE\$ = "TTE/001; 10 - Januari - 2022; AbsenUASMatematika; Alice".$$

Tahap pertama dalam proses verifikasi adalah menghitung nilai hash dari pesan M berdasarkan persamaan (14), hasilnya adalah:

$$h' = c67f50517185d171b8098fed309f982c.$$

Tabel 4. Hasil enkripsi plainteks

No	Blok m_i	No	Blok m_i
1	$c_1 = 099^{2299} \ mod \ 5609 = 1201$	17	$c_{17} = 098^{2299} \ mod \ 5609 = 151$
2	$c_2 = 054^{2299} \ mod \ 5609 = 4853$	18	$c_{18} = 056^{2299} \ mod \ 5609 = 4480$
3	$c_3 = 055^{2299} \ mod \ 5609 = 3531$	19	$c_{19} = 048^{2299} \ mod \ 5609 = 5299$
4	$c_4 = 102^{2299} \ mod \ 5609 = 1050$	20	$c_{20} = 057^{2299} \ mod \ 5609 = 3626$
5	$c_5 = 053^{2299} \ mod \ 5609 = 4494$	21	$c_{21} = 056^{2299} \ mod \ 5609 = 4480$
6	$c_6 = 048^{2299} \ mod \ 5609 = 5299$	22	$c_{22} = 102^{2299} \ mod \ 5609 = 1050$
7	$c_7 = 053^{2299} \ mod \ 5609 = 4494$	23	$c_{23} = 101^{2299} \ mod \ 5609 = 2931$
8	$c_8 = 049^{2299} \ mod \ 5609 = 5581$	24	$c_{24} = 100^{2299} \ mod \ 5609 = 1884$
9	$c_9 = 055^{2299} \ mod \ 5609 = 3531$	25	$c_{25} = 051^{2299} \ mod \ 5609 = 1514$
10	$c_{10} = 049^{2299} \ mod \ 5609 = 5581$	26	$c_{26} = 048^{2299} \ mod \ 5609 = 5299$
11	$c_{11} = 056^{2299} \ mod \ 5609 = 4480$	27	$c_{27} = 057^{2299} \ mod \ 5609 = 3626$
12	$c_{12} = 053^{2299} \ mod \ 5609 = 4494$	28	$c_{27} = 102^{2299} \ mod \ 5609 = 1050$
13	$c_{13} = 100^{2299} \ mod \ 5609 = 1884$	29	$c_{29} = 057^{2299} \ mod \ 5609 = 3626$
14	$c_{14} = 049^{2299} \ mod \ 5609 = 5581$	30	$c_{30} = 056^{2299} \ mod \ 5609 = 4480$
15	$c_{15} = 055^{2299} \ mod \ 5609 = 3531$	31	$c_{31} = 050^{2299} \ mod \ 5609 = 505$
16	$c_{16} = 049^{2299} \ mod \ 5609 = 5581$	32	$c_{32} = 099^{2299} \ mod \ 5609 = 1201$

Gambar 5. TTE beserta IE

Tahap selanjutnya adalah TTE akan didekripsi dengan menggunakan kunci publik yang juga disertakan dalam TTE tersebut berdasarkan persamaan (13) untuk menghasilkan MD h. Proses dekripsi berbasis KKPS ini dilakukan dengan menggunakan kunci publik ($e = 19$, $n = 5609$), dimana setiap blok chipertext c_i didekripsikan kembali menjadi blok-blok m_i. Disebabkan proses deskripsi menggunakan kunci publik maka persamaan (10) menjadi:

$$m_i = c_i^e \, (mod \ n) \, . \tag{16}$$

Pada Gambar 5, telah diperlihatkan TTE, dimana TTE = chiperteks, maka:

chiperteks = 1201 4853 3531 1050 4494 5299 4494 5581 3531 5581 4480 4494 1884 5581 3531 5581 151 4480 5299 3626 4480 1050 2931 1884 1514 5299 3626 1050 3626 4480 505 1201.

Proses deskripsi tiap blok-blok chiperteks dengan menggunakan persamaan (16), dapat dilihat pada Tabel 5, sebagai berikut:

Jika blok-blok m_i ini digabungkan, maka hasilnya adalah:

M=099 054 055 102 053 048 053 049 055 049 056 053 100 049 055 049 098 056 048 057056 102 101 100 051 048 057 102 057 056 050 099

Bisa dikatakan bahwa M merupakan hasil dekripsi dari TTE berdasarkan persamaan (13), jika M ini dikonversi kedalam sistem desimal pengkodean ASCII, maka hasilnya adalah nilai hash IE dari TTE, yaitu:

$$h = c67f50517185d171b8098fed309f982c$$

TABEL 5. Hasil enkripsi plainteks

No	Blok c_i	No	Blok c_i
1	$m_1 = 1201^{19}\ mod\ 5609 = 099$	17	$m_{17} = 151^{19}\ mod\ 5609 = 098$
2	$m_2 = 4853^{19}\ mod\ 5609 = 054$	18	$m_{18} = 4480^{19}\ mod\ 5609 = 056$
3	$m_3 = 3531^{19}\ mod\ 5609 = 055$	19	$m_{19} = 5299^{19}\ mod\ 5609 = 048$
4	$m_4 = 1050^{19}\ mod\ 5609 = 102$	20	$m_{20} = 3626^{19}\ mod\ 5609 = 057$
5	$m_5 = 4494^{19}\ mod\ 5609 = 053$	21	$m_{21} = 4480^{19}\ mod\ 5609 = 056$
6	$m_6 = 5299^{19}\ mod\ 5609 = 048$	22	$m_{22} = 1050^{19}\ mod\ 5609 = 102$
7	$m_7 = 4494^{19}\ mod\ 5609 = 053$	23	$m_{23} = 2931^{19}\ mod\ 5609 = 101$
8	$m_8 = 5581^{19}\ mod\ 5609 = 049$	24	$m_{24} = 1884^{19}\ mod\ 5609 = 100$
9	$m_9 = 3531^{19}\ mod\ 5609 = 055$	25	$m_{25} = 1514^{19}\ mod\ 5609 = 051$
10	$m_{10} = 5581^{19}\ mod\ 5609 = 049$	26	$m_{26} = 5299^{19}\ mod\ 5609 = 048$
11	$m_{11} = 4480^{19}\ mod\ 5609 = 056$	27	$m_{27} = 3626^{19}\ mod\ 5609 = 057$
12	$m_{12} = 4494^{19}\ mod\ 5609 = 053$	28	$m_{27} = 1050^{19}\ mod\ 5609 = 102$
13	$m_{13} = 1884^{19}\ mod\ 5609 = 100$	29	$m_{29} = 3626^{19}\ mod\ 5609 = 057$
14	$m_{14} = 5581^{19}\ mod\ 5609 = 049$	30	$m_{30} = 4480^{19}\ mod\ 5609 = 056$
15	$m_{15} = 3531^{19}\ mod\ 5609 = 055$	31	$m_{31} = 505^{19}\ mod\ 5609 = 050$
16	$m_{16} = 5581^{19}\ mod\ 5609 = 049$	32	$m_{32} = 1201^{19}\ mod\ 5609 = 099$

Jika membandingkan h dengan h' yang hasilnya adalah $h = h'$, berarti TTE yang diverifikasi adalah otentik, TTE tersebut asli dan bersumber dari penandatangan yang asli, dan penandatangan tersebut yang membuat kunci privat sekaligus kunci publik, sehingga penandatangan tidak bisa menyangkal telah menandatangan TTE tersebut.

Pada kenyataannya akan ada terjadi tindakan yang memanipulasi tanda tangan dengan mengubah IE dari TTE. Misalkan pada Gambar 6, akan diperlihatkan TTE yang IE nya telah dirubah, dimana judul diganti menjadi "Absen UTS Matematika", seperti:

GAMBAR 6. TTE dengan IE termodifikasi

Dari Gambar 6, dapat ditentukan IE yang melekat pada TTE tersebut adalah:

$$IE\$ = "TTE/001; 10 - Januari - 2022; AbsenUTSMatematika; Alice". \quad (17)$$

Tahap pertama adalah menghitung nilai hash dari pesan M, dimana M disini adalah struktur dari IE, berdasarkan persamaan (14) yang hasilnya adalah:

$$h = c3e0d87bf52fb6437bd8c5d885ed4087.$$

Tahap selanjutnya adalah TTE akan didekripsi dengan menggunakan kunci publik berdasarkan persamaan (13). Tahapan proses ini sama dengan proses dekripsi yang telah dilakukan sebelumnya. Pada proses dekripsi ini didapat hasil:

$$h = c67f50517185d171b8098fed309f982c.$$

Dengan membandingkan h dengan h' yang hasilnya adalah $h \neq h'$, berarti TTE yang diverifikasi tidak valid, IE dari TTE tersebut telah dimodifikasi, dan bukan berasal dari penandatangan yang asli. Proses penandatangan yang menggunakan algoritma enkripsi dan proses verifikasi yang menggunakan algoritma dekripsi, penulis implementasikan kedalam bahasa pemograman PHP versi 7 dan menggunakan sistem operasi berbasis 32-bit. Pada proses penandatangan TTE diperlukan waktu 0.024001836776733 detik. Sedangkan waktu yang diperlukan dalam proses verifikasi TTE ini adalah: 0.000000000000001 detik. Terlihat bahwa waktu proses deskripsi lebih kecil daripada waktu proses enkripsi, ini dikarenakan ukuran *digit* d lebih besar dari pada ukuran *digit* e yang dipilih. Perbandingan nilai waktu proses ini bisa berubah terbalik jika ukuran *digit* nilai e yang dipilih lebih besar daripada ukuran *digit* nilai d.

3.4. **Keamanan.** Verifikasi TTE dengan teknik otentikasi berbasis KKPS menggunakan algoritma kriptografi RSA yang keamanannya terletak sulitnya proses memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Algoritma kriptografi RSA aman jika ukuran *digit* n cukup besar. Berikut pada Tabel 6, akan diperlihatkan ukuran *digit* n, jumlah operasi yang diperlukan untuk memfaktorkan n, dan waktu yang diperlukan pada proses memfaktorkan n, dengan metode Schroeppel yang dipublikasi oleh R.L. Rivest, dkk [16].

Tabel 6. Waktu proses memfaktorkan n berdasarkan ukuran *digit* n

Digits	Jumlah Operasi	Waktu
50	1.4×10^{10}	3.9 Jam
75	9.0×10^{12}	104 Hari
100	2.3×10^{15}	74 Tahun
200	1.2×10^{23}	3.8×10^{9} Tahun
300	1.5×10^{29}	4.9×10^{15} Tahun
500	1.3×10^{39}	4.2×10^{15} Tahun

Dari Tabel 6, dapat disimpulkan bahwa n dengan ukuran diatas 100 *digit* memberikan keamanan moderat terhadap serangan menggunakan teknologi komputer saat ini, dan menggunakan n dengan ukuran 200 *digit* keatas akan memberikan margin keamanan yang tinggi terhadap serangan dengan menggunakan teknologi komputer masa depan yang semakin canggih, dimana waktu yang dibutuhkan dalam proses memfaktorkan n menjadi faktor-faktor prima membutuhkan waktu 3.8×10^{9} tahun, bahkan memerlukan waktu 4.2×10^{15} tahun jika n berukuran 500 *digit*.

4. Simpulan

Berdasarkan hasil dari penelitian ini, maka dapat ditarik simpulan, bahwa verifikasi TTE dengan teknik otentikasi berbasis KKPS menggunakan algoritma kriptografi RSA merupakan metode yang sangat tepat digunakan untuk menjamin keaslian suatu TTE dan IE yang melekat pada TTE tersebut.

Verifikasi TTE dengan teknik otentikasi berbasis KKPS menggunakan algoritma kriptografi RSA merupakan anti-penyangkalan atau dapat menghindari adanya penyangkalan bahwa seseorang telah menandatangani TTE dengan IE yang melekat pada TTE tersebut.

Proses verifikasi tanda tangan dengan menggunakan teknik otentikasi berbasis KKPS menggunakan algoritma kriptografi RSA, dapat dilakukan oleh semua pihak, tanpa diperlukan pihak ke tiga untuk memverifikasi TTE secara khusus.

TTE berbasis KKPS dengan menggunakan algoritma kriptografi RSA akan sangat sulit untuk dipalsukan, disebabkan berasosiasi dengan kombinasi IE dan sepasang kunci privat dan publik secara unik.

Waktu proses penandatangan dan verifikasi sangat tergantung dari ukuran digit kunci publik dan kunci privatnya. Dengan membandingkan d dan e, jika ukuran digit $d > e$, maka waktu penandatangan lebih lama daripada waktu verfikasi, begitu juga sebaliknya.

TTE berbasis KKPS dengan menggunakan algoritma kriptografi RSA dikatakan sangat aman jika n 200 *digit* keatas. Hal ini disebabkan proses memfaktorkan bilangan 200 *digit* menjadi faktor-faktor prima membutuhkan waktu proses komputasi selama 3.8×10^9 tahun.

Daftar Pustaka

[1] Alawiyah, M., Kusuma, D. A., and Ruchjana, B. N., 2021, Application of Generalized Space Time Autoregressive Integrated (GSTARI) model in the phenomenon of covid-19, *J. Phys. Conf. Ser.*, 1722(1), p. 12035, doi: 10.1088/1742-6596/1722/1/012035.

[2] Darmawan, G., Rosadi, D., Ruchjana, B., 2021 Covid-19 daily forecasting during ramadhan in countries with high muslim population, *J. Phys. Conf. Ser.*, 1722, p. 12092.

[3] Pontoh, R. S., Zahroh, S., Hidayat, Y., Aldella, R., Jiwani, N.M., and Sukono, 2020, Covid-19 Modelling in South Korea using A Time Series Approach, *Int. J. Adv. Sci. Technol.*, 29(7 SE - Articles), pp. 1620-1632.

[4] Jamaludin, S., Azmir, N. A., Mohamad Ayob, A. F., and Zainal, N., 2020, COVID-19 exit strategy: Transitioning towards a new normal, *Ann. Med. Surg.*, 59, pp. 165170, doi: https://doi.org/10.1016/j.amsu.2020.09.046.

[5] Perdana, T., Chaerani, D., Achmad, A. L. H., and Hermiatin, F. R., 2020, Scenarios for handling the impact of COVID-19 based on food supply network through regional food hubs under uncertainty, *Heliyon*, 6(10), p. e05128, doi: https://doi.org/10.1016/j.heliyon.2020.e05128.

[6] Syuhada, K., Wibisono, A., Hakim, A., and Addini, F., 2021, Covid-19 risk data during lockdown-like policy in Indonesia, *Data Br.*, 35, p. 106801, doi: https://doi.org/10.1016/j.dib.2021.106801.

[7] Afrianty, T. W., Artatanaya, I. G., and Burgess, J., 2021, Working from home effectiveness during Covid-19: Evidence from university staff in Indonesia, *Asia Pacific Manag. Rev.*, doi: https://doi.org/10.1016/j.apmrv.2021.05.002.

[8] Ayu, S. and Lahmi, A., 2020, Peran e-commerce terhadap perekonomian Indonesia selama pandemi Covid-19, *J. Kaji. Manaj. Bisnis*, 9(2), pp. 114123, doi: https://doi.org/10.24036/jkmb.10994100.

[9] Aulia, S., 2020, Pola Perilaku Konsumen Digital Dalam Memanfaatkan Aplikasi Dompet Digital, *J. Komun.*, 12(2), pp. 311324, doi: http://dx.doi.org/10.24912/jk.v12i2.9829.

[10] Yusuf, 2020, Pandemi Picu Peningkatan Transaksi Online, Kominfo Siapkan Empat Kebijakan Percepat Digitalisasi, *Kementerian Komunikasi dan Informatika*.

[11] Schneier, B., 1996, *Applied cryptography : protocols, algorithms, and source code in C*, Second edition, New York : Wiley.

[12] Kominfo, 2012, *Penyelenggaraan Sistem Dan Transaksi Elektronik*, Indonesia.

[13] Muhammad Andalan, A., 2019, Kedudukan Tanda Tangan Elektronik dalam Transaksi Teknologi Finansial, *Jurist-Diction*, 2(6), pp. 19311950, doi: http://dx.doi.org/10.20473/jd.v2i6.15921.

[14] Isnaini, H. F. and Karyati, K., 2017, Penerapan skema tanda tangan Schnorr pada pembuatan tanda tangan digital, *Pythagoras J. Pendidik. Mat.*, 12(1), doi: 10.21831/pg.v12i1.11631.

[15] DPR-RI, 2008, *Undang Undang Republik Indonesia Nomor 11 Tahun 2008*.

[16] Rivest, R. L., Shamir, A., and Adleman, L., 1978, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Commun. ACM*, 21(2), pp. 120126, doi: 10.1145/359340.359342.

[17] Azdy, R., 2016, Tanda tangan Digital Menggunakan Algoritme Keccak dan RSA, *J. Nas. Tek. Elektro dan Teknol. Inf.*, 5, doi: 10.22146/jnteti.v5i3.255.

[18] Vollala, S., Varadhan, V. V, Geetha, K., and Ramasubramanian, N., 2014, Efficient modular multiplication algorithms for public key cryptography, in *2014 IEEE International Advance Computing Conference (IACC)*, pp. 7478, doi: 10.1109/IAdCC.2014.6779297.

[19] Stallings, W., 2017, *Cryptography and Network Security Principles and Practices*, Seventh Ed., Upper Saddle River, N.J: Pearson Education Limited.

[20] Mumtaz M., and Ping, L., 2019, Forty years of attacks on the RSA cryptosystem: A brief survey, *J. Discret. Math. Sci. Cryptogr.*, 22(1), pp. 929, doi: 10.1080/09720529.2018.1564201.

[21] Menezes, A. J., Vanstone, S. A., and Van Oorschot, P. C., 1996, *Handbook of Applied Cryptography*, 1st ed., USA: CRC Press, Inc.

[22] Sharma, A. and Mittal, S.K., 2018, Attacks on Cryptographic Hash Functions and Advances, 5, pp. 8996.

[23] Melina, Putra, E. K., Witanti, W., Sukrido, and Kusumaningtyas, V. A., 2020, Design and Implementation of Multi Knowledge Base Expert System Using the SQL Inference Mechanism for Herbal Medicine, *J. Phys. Conf. Ser.*, 1477(2),doi: 10.1088/1742-6596/1477/2/022007.

[24] Stinson, D. R. and Paterson, M. B., 2002, *Cryptography: Theory and Practice,Second Edition*, Fourth Ed., CRC Press, Inc.

[25] Noroozi, E., Daud, S., and Sabouhi, A., 2013, Secure Digital Signature Schemes Based on Hash Functions.

[26] Official_php, 2001, Function Hash, *Documentation Manual Php*.

[27] Ardy, R. D., Indriani, O. R., Sari, C. A., Setiadi, D. R. I. M., and Rachmawanto, E. H., 2017, Digital image signature using triple protection cryptosystem (RSA, Vigenere, and MD5), in *2017 International*

Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), pp. 8792, doi: 10.1109/ICON-SONICS.2017.8267827.

[28] Aufa, F. J., Endroyono, and Affandi, A., 2018, Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm, in *2018 4th International Conference on Science and Technology (ICST)*, pp. 15, doi: 10.1109/ICSTC.2018.8528584.

[29] Somsuk, K. and Thakong, M., 2020, Authentication system for e-certificate by using RSA's digital signature, *TELKOMNIKA*, 18, p. 2948, doi: 10.12928/telkomnika.v18i6.17278.

[30] Atmaja, I. M. A. D. S., Astawa, I. N. G. A., Wisswani, N. W., Nugroho, I. M. R. A., Sunu, P. W., and Wiratama, I. K., 2020, Document Encryption Through Asymmetric RSA Cryptography, in *2020 International Conference on Applied Science and Technology (iCAST)*, pp. 4649, doi: 10.1109/iCAST51016.2020.9557723.

[31] Sharif, A., Ginting, D. S., and Dias, A. D., Securing the Integrity of PDF Files using RSA Digital Signature and SHA-3 Hash Function, 2021, in *2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pp. 154159, doi: 10.1109/DATABIA53375.2021.9650121.

[32] Munir, R., 2019, *Kriptografi*, Second Ed., Bandung: Informatika.

[33] Baumslag, G., Fine, B., Kreuzer, M., and Rosenberger, G., 2015, *A Course in Mathematical Cryptography*, Hamburg: De Gruyter.