

IMPLEMENTASI *DIGITAL SIGNATURE* MENGGUNAKAN SHA-256 DAN RSA UNTUK AUTENTIKASI PESAN

Dameria Naomi Simatupang, Vanyariska Indriani, Marzuqi Rouf, Eka Ardhianto, Aji Supriyanto

Teknologi Informasi, Universitas Stikubank Semarang
Jl. Tri Lomba Juang, Mugassari Semarang, Indonesia
damerianaomi0016@mhs.unisbank.ac.id

ABSTRAK

Perkembangan komunikasi *digital* meningkatkan kebutuhan akan mekanisme keamanan yang mampu menjamin keaslian dan integritas data selama proses pertukaran informasi. Namun, penggunaan jaringan internet yang bersifat terbuka menimbulkan risiko manipulasi data dan serangan *Man in the Middle* (MitM), sehingga diperlukan metode autentikasi yang andal. Penelitian ini bertujuan mengimplementasikan dan mengevaluasi *digital signature* menggunakan kombinasi algoritma SHA-256 dan RSA untuk menjaga autentikasi serta integritas pesan pada sistem berbasis *web*. Metode yang digunakan adalah pendekatan eksperimental melalui perancangan dan implementasi sistem menggunakan PHP dan *library OpenSSL*. Proses *digital signature* dilakukan dengan menghasilkan nilai *hash* menggunakan SHA-256, kemudian menandatangani menggunakan *private key* RSA dan memverifikasinya menggunakan *public key*. Pengujian dilakukan pada tiga skenario, yaitu pesan asli, modifikasi pesan, dan manipulasi *signature*, serta pengukuran waktu proses *signing* dan *verification*. Hasil penelitian menunjukkan bahwa sistem berhasil memverifikasi pesan asli dengan status *valid*, sedangkan perubahan isi pesan maupun manipulasi *signature* menyebabkan verifikasi gagal. Selain itu, waktu proses *signing* pada RSA 2048-bit tercatat sekitar 0,003 detik dan *verification* sekitar 0,0002 detik. Hasil tersebut menunjukkan bahwa kombinasi SHA-256 dan RSA efektif dalam menjaga autentikasi dan integritas pesan pada komunikasi *digital* berbasis *web*.

Kata kunci : autentikasi pesan, *digital signature*, integritas data, kriptografi, RSA, SHA-256

1. PENDAHULUAN

Perkembangan teknologi yang pesat mendorong transformasi *digital* dari proses konvensional menjadi otomatis, termasuk perubahan dokumen fisik dan tanda tangan basah menjadi dokumen serta tanda tangan *digital* yang lebih mudah, cepat, dan tidak terbatas oleh jarak maupun waktu [1]. Di sisi lain, penggunaan internet sebagai media komunikasi yang terbuka dan dapat diakses secara luas menjadikannya rentan terhadap berbagai ancaman keamanan, seperti penyadapan dan manipulasi data oleh pihak yang tidak bertanggung jawab [2]. Terlebih lagi, proses transmisi data pada jaringan terbuka berpotensi disusupi oleh pihak ketiga melalui serangan *Man in the Middle* (MitM), yang memungkinkan terjadinya perubahan informasi tanpa sepengetahuan pengirim maupun penerima [3]. Kondisi tersebut meningkatkan kebutuhan akan mekanisme yang mampu menjaga kerahasiaan, integritas, serta keaslian informasi dalam pertukaran data elektronik [4]. Penerapan mekanisme keamanan tersebut juga sejalan dengan prinsip keamanan informasi yang menitikberatkan pada aspek *Confidentiality*, *Integrity*, dan *Availability* (CIA Triad) [5].

Salah satu ancaman yang sering terjadi dalam komunikasi *digital* adalah serangan *Man in the Middle* (MitM), yaitu kondisi ketika pihak ketiga dapat menyusup ke dalam proses komunikasi dan memodifikasi isi pesan tanpa diketahui oleh pengguna [6]. Serangan semacam ini dapat menyebabkan berkurangnya kepercayaan terhadap informasi yang dipertukarkan karena tidak adanya jaminan bahwa data

yang diterima masih sama dengan data yang dikirimkan. Oleh karena itu, diperlukan penerapan teknik kriptografi yang mampu melindungi integritas dokumen *digital*, khususnya pada data yang bersifat sensitif [7]. Pendekatan kriptografi yang tepat terbukti mampu meningkatkan tingkat keamanan dokumen *digital* dari modifikasi oleh pihak yang tidak berwenang [8].

Salah satu pendekatan yang banyak digunakan adalah *digital signature*, yang berfungsi untuk menjamin integritas data, autentikasi pengirim, serta mencegah penyangkalan (*non-repudiation*) melalui mekanisme kriptografi kunci publik [9]. Dengan adanya *digital signature*, penerima dapat memverifikasi bahwa pesan benar – benar berasal dari pengirim yang sah dan tidak mengalami perubahan selama proses transmisi. Dalam implementasinya, *digital signature* umumnya memanfaatkan fungsi *hash* untuk menghasilkan *message digest* sebagai representasi unik dari suatu pesan. Penelitian ini menggunakan algoritma *Secure Hash Algorithm-256* (SHA-256) untuk menghasilkan nilai *hash* yang aman dan tahan terhadap *collision* [10]. Selain itu, algoritma RSA digunakan sebagai mekanisme kriptografi kunci publik dalam proses penandatanganan dan verifikasi, sebagaimana juga diterapkan dalam protokol keamanan seperti *Transport Layer Security* (TLS) [11]. Kombinasi SHA-256 dan RSA dipilih karena mampu mendukung proses autentikasi serta menjaga integritas data melalui mekanisme *hashing* dan penandatanganan *digital* yang saling melengkapi.

Meskipun berbagai penelitian telah mengimplementasikan *digital signature* menggunakan algoritma RSA dan fungsi *hash*, sebagian besar penelitian masih berfokus pada proses dasar penandatanganan dan verifikasi tanpa mengevaluasi kemampuan sistem dalam mendeteksi berbagai bentuk manipulasi data secara sistematis. Pengujian yang dilakukan umumnya terbatas pada validasi tanda tangan dalam kondisi normal, sehingga belum memberikan gambaran yang komprehensif mengenai ketahanan sistem terhadap perubahan isi pesan maupun manipulasi *signature*. Selain itu, implementasi berbasis *web* yang memanfaatkan *library OpenSSL* serta menyediakan fleksibilitas pemilihan ukuran kunci RSA masih relatif terbatas, khususnya dalam konteks analisis efisiensi waktu proses *signing* dan *verification*.

Berdasarkan permasalahan tersebut, penelitian ini berfokus pada implementasi *digital signature* menggunakan kombinasi algoritma SHA-256 dan RSA dalam sebuah sistem berbasis *web*. Sistem yang dibangun tidak hanya melakukan proses *signing* dan *verification*, tetapi juga mampu mendeteksi perubahan pesan dan manipulasi tanda tangan *digital* melalui beberapa skenario pengujian, yaitu pesan asli, modifikasi pesan, dan manipulasi *signature*. Selain itu, penelitian ini turut mengevaluasi efisiensi sistem berdasarkan waktu proses *signing* dan *verification*. Dengan demikian, penelitian ini bertujuan untuk menganalisis efektivitas dan keandalan kombinasi algoritma SHA-256 dan RSA dalam menjaga autentikasi serta integritas pesan pada komunikasi *digital* berbasis *web*.

2. TINJAUAN PUSTAKA

Digital signature atau tanda tangan digital merupakan salah satu mekanisme keamanan informasi yang bertujuan untuk menjamin autentikasi komponen pengiriman dan menjaga integritas data dari risiko manipulasi selama proses transmisi. Penelitian sebelumnya telah mengembangkan berbagai metode untuk mengoptimalkan keandalan tanda tangan digital menggunakan kombinasi fungsi *hash* dan kriptografi kunci publik. Hutagalung [21] mengimplementasikan tanda tangan digital menggunakan kombinasi algoritma SHA-256 dan RSA untuk mengamankan data teks. Hasil penelitian menunjukkan bahwa integrasi kedua algoritma tersebut menghasilkan pertahanan yang kuat dari risiko modifikasi ilegal karena sifat *collision-resistance* pada SHA-256 dan kompleksitas matematis faktorisasi pada RSA.

Penelitian lainnya oleh Nurjaman [1] menerapkan sistem penandatanganan dokumen digital menggunakan standar *hashing* terbaru SHA3-512 dikombinasikan dengan RSA pada sistem penyimpanan berkas. Hasilnya, mekanisme tersebut sukses mempertahankan keaslian dokumen dari ancaman manipulasi siber, meskipun membutuhkan sumber daya komputasi yang selaras dengan panjangnya nilai *digest* yang dihasilkan. Sementara

itu, Az-Zahra [28] mengembangkan sistem autentikasi dokumen digital dengan mengintegrasikan algoritma SHA-256 dan RSA ke dalam bentuk *QR Code*. Sistem ini terbukti 100% akurat dalam mendeteksi pemalsuan dokumen serta memberikan kemudahan aksesibilitas verifikasi fisik-ke-digital secara instan melalui pemindaian kode.

Berdasarkan penelitian-penelitian sebelumnya, kombinasi algoritma *hashing* dan kriptografi asimetris terbukti sangat efektif untuk mengamankan integritas data berkas maupun dokumen. Namun, penelitian lebih lanjut diperlukan untuk menguji efisiensi komputasi dari segi waktu proses ketika diimplementasikan pada komunikasi pesan teks *real-time* berbasis *web* guna menangkal serangan *Man-in-the-Middle* (MitM). Oleh karena itu, penelitian ini mengkaji penerapan algoritma SHA-256 dan RSA (2048-bit) menggunakan bahasa pemrograman PHP dan *library OpenSSL* dengan mengukur performa kecepatan proses *signing* dan *verification* secara komprehensif.

2.1. Keamanan Informasi

Keamanan Informasi merupakan upaya terstruktur untuk melindungi data, informasi, dan sistem dari berbagai ancaman seperti akses ilegal, manipulasi, hingga kerusakan infrastruktur. Di era transformasi digital, urgensi keamanan informasi meningkat tajam karena mayoritas pertukaran data dilakukan melalui jaringan internet terbuka yang rentan terhadap *malware*, penyadapan, dan serangan siber lainnya. Keamanan informasi bertujuan melindungi aset organisasi melalui penerapan sistem dan teknologi informasi yang tepat. Sistem yang andal tidak hanya meminimalkan risiko kerentanan digital, tetapi juga meningkatkan kepercayaan (*trust*) pengguna karena adanya jaminan perlindungan data dari manipulasi maupun pencurian [12].

Pilar utama domain ini dikenal sebagai *CIA Triad*, yang terdiri dari *Confidentiality* (kerahasiaan akses pihak berwenang), *Integrity* (pemeliharaan keutuhan data selama penyimpanan dan transmisi), serta *Availability* (ketersediaan sistem saat dibutuhkan pengguna) [12]. Selain ketiga pilar tersebut, arsitektur keamanan modern juga mencakup aspek *authentication* untuk memvalidasi keaslian identitas pengguna, serta *non-repudiation* guna mencegah penyangkalan transaksi yang telah dilakukan [13]. Untuk menghadapi ancaman siber yang terus berevolusi, integrasi berbagai teknologi seperti kriptografi, fungsi *hashing*, tanda tangan digital (*digital signature*), dan *firewall* menjadi solusi krusial dalam memproteksi ekosistem sistem informasi modern [14].

2.2. Kriptografi Asimetris (Asymmetric Cryptography)

Kriptografi Asimetris (*Asymmetric Cryptography*) adalah metode yang menggunakan sepasang kunci terikat secara matematis, yaitu kunci

publik (*public key* untuk enkripsi/verifikasi) dan kunci privat (*private key* untuk dekripsi/signing). Metode ini menawarkan tingkat keamanan yang lebih tinggi daripada kriptografi simetris karena kunci privat sepenuhnya dirahasiakan oleh pemiliknya [15]. Kriptografi asimetris banyak digunakan karena mampu menyediakan aspek *confidentiality*, *authentication*, *integrity*, dan *non-repudiation* secara simultan dalam komunikasi internet [16].

Algoritma asimetris seperti *Elliptic Curve Cryptography* (ECC) dan RSA merupakan landasan utama keamanan digital kontemporer, yang diimplementasikan secara luas pada SSL/TLS, VPN, enkripsi surel, serta sertifikat digital [17]. Kekuatan keamanan algoritma ini sangat bergantung pada kompleksitas matematika; misalnya pada RSA yang bertumpu pada tingginya tingkat kesulitan pemfaktoran dua bilangan prima besar, di mana semakin besar ukuran kunci maka semakin tinggi keamanannya meskipun memerlukan waktu komputasi yang lebih lama [13]. Dalam praktiknya, teknologi ini sering dikombinasikan dengan fungsi *hash* untuk mengoptimalkan efisiensi autentikasi data, seperti pada mekanisme *digital signature* yang mengubah pesan menjadi *message digest* sebelum ditandatangani oleh kunci privat [18]. Kombinasi inilah yang menjadikannya teknologi vital pada aplikasi transaksi elektronik, komunikasi data, dan dokumen digital berbasis internet saat ini.

2.3. Digital Signature

Digital Signature (tanda tangan digital) merupakan mekanisme kriptografi kunci publik yang menggunakan sepasang kunci, yaitu kunci privat (*private key*) milik pengirim untuk menandai dokumen dan kunci publik (*public key*) untuk proses verifikasi oleh penerima [19]. Fungsi utama teknologi ini adalah menjamin bahwa dokumen elektronik benar-benar otentik berasal dari pengirim yang sah, bebas dari modifikasi selama transmisi, serta mendukung aspek *non-repudiation* (anti-penyangkalan) [20].

Secara teknis, proses diawali dengan mengubah pesan menjadi *message digest* berukuran tetap melalui fungsi *hashing* seperti SHA-256, yang kemudian dienkripsi menggunakan kunci privat untuk menghasilkan tanda tangan digital. Penerima cukup membandingkan *hash* baru dari pesan dengan hasil dekripsi tanda tangan menggunakan kunci publik untuk menguji keabsahannya [21]. Integrasi SHA-256 dan RSA terbukti meningkatkan keamanan secara signifikan karena sifatnya yang sangat sensitif terhadap perubahan data, di mana manipulasi sekecil apa pun otomatis akan menggagalkan proses verifikasi [18]. Berkat kapabilitasnya tersebut, *digital signature* kini diimplementasikan secara luas pada sistem *e-government*, transaksi elektronik, *email security*, dan dokumen berbasis web sebagai komponen vital dalam sistem keamanan informasi modern [19].

2.4. SHA-256

Secure Hash Algorithm-256 (SHA-256) merupakan fungsi *hash* kriptografi satu arah dalam rumpun SHA-2 yang menghasilkan nilai *hash* (*message digest*) unik sepanjang 256 bit dari suatu data atau pesan, sehingga mustahil dikembalikan menjadi data asli. Menurut Halimi et al. (2024) [22], SHA-256 menawarkan tingkat keamanan tinggi berkat ketahanannya terhadap *collision attack* (dua input berbeda menghasilkan *hash* serupa) dan *preimage attack* (menemukan data asli berdasarkan nilai *hash*). Kemampuannya dalam menjaga integritas data diperkuat oleh karakteristik *avalanche effect*, di mana modifikasi pada satu karakter pesan saja akan memicu perubahan nilai *hash* yang sangat kontras [14].

Dalam aplikasinya pada mekanisme *digital signature*, penggunaan SHA-256 terbukti meningkatkan sensitivitas sistem terhadap manipulasi data. Jika suatu pesan diubah, nilai *hash* perhitungan ulang otomatis berbeda dan menggagalkan proses verifikasi. Dibandingkan algoritma *hash* pendahulunya seperti SHA-1, SHA-256 memberikan proteksi yang jauh lebih andal, sehingga kini diimplementasikan secara luas pada sistem keamanan informasi modern termasuk autentikasi kata sandi, *blockchain*, SSL/TLS, token keamanan, dan sertifikat digital [22].

2.5. RSA

RSA *Cryptosystem* merupakan algoritma kriptografi kunci publik berbasis asimetris yang dikembangkan oleh Ronald Rivest, Adi Shamir, dan Leonard Adleman, serta digunakan secara luas berkat tingkat keamanannya yang tinggi [15]. RSA beroperasi menggunakan sepasang kunci, di mana kunci publik (*public key*) berfungsi untuk enkripsi atau verifikasi, sedangkan kunci privat (*private key*) digunakan untuk dekripsi atau penandatanganan (*signing*). Kekuatan keamanan RSA bertumpu penuh pada tingkat kesulitan matematika dalam memfaktorkan dua bilangan prima berukuran besar yang membentuk modulus sistem tersebut.

Proses pembangkitan kunci RSA diawali dengan pemilihan dua bilangan prima besar untuk menghasilkan nilai modulus serta pasangan kunci publik dan privat, di mana peningkatan ukuran kunci akan memperkuat proteksi sistem secara signifikan. Dalam domain komunikasi digital, RSA menjadi fondasi vital untuk memelihara aspek autentikasi dan integritas data, khususnya pada mekanisme tanda tangan digital (*digital signature*) lewat skema enkripsi nilai *hash* menggunakan kunci privat yang dilanjutkan dengan verifikasi melalui kunci publik.

Kendati memerlukan beban komputasi yang lebih besar pada ukuran kunci standar seperti 2048-bit atau 4096-bit dibandingkan dengan algoritma simetris, RSA tetap menjadi solusi efektif yang diimplementasikan secara luas pada SSL/TLS, VPN, enkripsi surel, sertifikat digital, hingga autentikasi jaringan. Guna mengatasi kendala performa tersebut,

RSA lazim dikombinasikan dengan fungsi *hash* seperti SHA-256 agar proses *signing* dan *verification* dapat berjalan dengan jauh lebih efisien.

3. METODE PENELITIAN

3.1. Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimental yang bertujuan untuk mengimplementasikan serta mengevaluasi mekanisme *digital signature* dalam menjamin integritas dan autentikasi pesan. Tahapan penelitian dilakukan secara sistematis, meliputi studi literatur, analisis algoritma, perancangan sistem, implementasi, pengujian, serta analisis hasil.

Pada tahap studi literatur dilakukan pengkajian terhadap konsep kriptografi, *digital signature*, algoritma RSA, serta fungsi *hash* SHA-256 dari berbagai sumber ilmiah. Tahap ini tidak hanya digunakan sebagai landasan teori, tetapi juga untuk menentukan metode yang paling sesuai dalam membangun sistem yang mampu mendeteksi manipulasi data secara efektif.

Tahap selanjutnya adalah perancangan dan implementasi sistem *digital signature* berbasis *web* menggunakan PHP dan *library OpenSSL*. Sistem dirancang untuk melakukan proses penandatanganan (*signing*) dan verifikasi (*verification*), yang kemudian diuji melalui beberapa skenario untuk mengevaluasi akurasi dan performa sistem.

3.2. Mekanisme *Digital Signature* Berbasis SHA-256 dan RSA

Tanda tangan *digital* digunakan sebagai mekanisme untuk menjamin keaslian dan integritas pesan dalam komunikasi *digital*. Sejak diperkenalkan pada tahun 1991, *digital signature* telah berkembang sebagai metode autentikasi yang memungkinkan pengirim menambahkan kode unik yang hanya dapat diverifikasi menggunakan kunci tertentu [23]. Dalam implementasinya, *digital signature* memanfaatkan algoritma kriptografi asimetris untuk memastikan bahwa pesan tidak dapat diubah tanpa terdeteksi [24].

Dalam penelitian ini, mekanisme *digital signature* dibangun dengan mengombinasikan fungsi *hash* SHA-256 dan algoritma RSA. Integrasi kedua algoritma ini bertujuan untuk menghasilkan sistem yang tidak hanya mampu melakukan autentikasi, tetapi juga memiliki sensitivitas tinggi terhadap perubahan data.

Proses pertama adalah *hashing* menggunakan SHA-256. Algoritma SHA dirancang sebagai fungsi kriptografi untuk mendukung keamanan data melalui transformasi menggunakan fungsi *hash* [25]. Dalam penelitian ini, SHA-256 digunakan untuk mengubah pesan teks menjadi nilai *hash* dengan panjang tetap 256 bit. Pemilihan SHA-256 didasarkan pada ketahanannya terhadap serangan kolisi [26], sehingga perubahan kecil pada pesan akan menghasilkan perbedaan nilai *hash* yang signifikan. Secara

matematis, proses ini dinyatakan sebagai pada persamaan (1).

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

Keterangan:

M : pesan asli (*Message*)

h : fungsi *hash* SHA-256

H : nilai *hash* (*Message digest*) hasil dari proses *hashing*

Selanjutnya, nilai *hash* yang dihasilkan akan ditandatangani menggunakan algoritma RSA. RSA merupakan algoritma kunci publik yang banyak digunakan karena tingkat keamanannya yang tinggi, yang bergantung pada kesulitan pemfaktoran bilangan prima besar [27]. Parameter utama dalam RSA meliputi p, q, n, $\phi(n)$, e, dan d, yang digunakan dalam proses pembangkitan kunci [21].

Proses pembangkitan parameter RSA secara umum dinyatakan pada persamaan (2).

$$n = p \times q \\ \Phi(n) = (p - 1)(q - 1) \quad (2)$$

Keterangan:

p, q : bilangan prima besar

n : modulus RSA

$\phi(n)$: fungsi totien Euler

Dalam penelitian ini, proses *signing* dilakukan dengan mengenkripsi nilai *hash* menggunakan *private Key* RSA, sehingga menghasilkan *digital signature* pada persamaan (3).

$$S = H^d \text{ mod } n \quad (3)$$

Keterangan :

S : *digital signature*

H : nilai *hash* pesan

d : kunci privat (*private Key*)

n : modulus RSA

Pendekatan ini memastikan bahwa hanya pemilik *private Key* yang dapat menghasilkan *signature* yang *valid*.

Pada sisi penerima, proses verifikasi dilakukan dengan menghitung ulang nilai *hash* dari pesan dan mendekripsi *signature* menggunakan *public Key* RSA. RSA sebagai algoritma kriptografi asimetris yang berbasis pada *Integer Factorization Problem* (IFP) masih banyak digunakan karena keamanannya yang tinggi [28]. Proses verifikasi dinyatakan pada persamaan (4).

$$H' = h(M), H'' = S^e \text{ mod } n \quad (4)$$

Keterangan:

H' : *hash* hasil perhitungan ulang dari pesan

H'' : *hash* hasil dekripsi *signature*

S : *digital signature*

e : kunci publik (*public Key*)

n : modulus RSA

Pesan dinyatakan *valid* apabila persamaan (5) terpenuhi.

$$H' = H'' \quad (5)$$

Jika terjadi perubahan pada pesan atau manipulasi terhadap *signature*, maka kedua nilai tersebut tidak akan sama. Hal ini menunjukkan bahwa

kombinasi SHA-256 dan RSA tidak hanya berfungsi sebagai mekanisme autentikasi, tetapi juga sebagai alat deteksi integritas data yang efektif.

3.3. Metode yang Diusulkan

Metode yang diusulkan dalam penelitian ini adalah implementasi sistem *digital signature* berbasis *web* yang mengintegrasikan algoritma SHA-256 dan RSA untuk menjamin integritas dan autentikasi pesan.

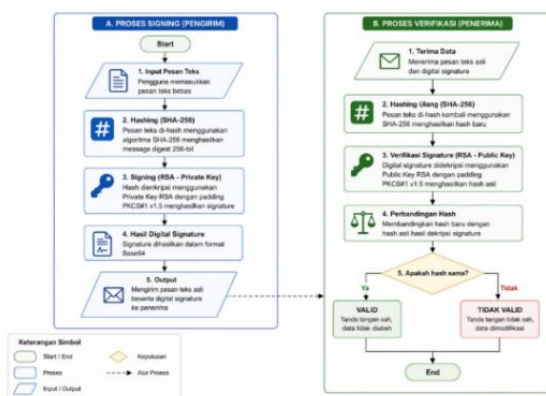
Berbeda dengan pendekatan dasar yang hanya melakukan validasi tanda tangan, metode ini menekankan pada kemampuan sistem dalam mendeteksi perubahan data melalui perbandingan nilai *hash* serta mengevaluasi efisiensi proses kriptografi berdasarkan waktu eksekusi. Proses dalam sistem terdiri dari beberapa langkah berikut.

- Input pesan teks oleh pengguna
- Proses *hashing* menggunakan SHA-256
- Proses *signing* menggunakan *private Key* RSA
- Pengiriman pesan beserta *digital signature*
- Proses *verification* menggunakan *public Key* RSA
- Validasi hasil berdasarkan kesesuaian *hash*

Pendekatan ini memungkinkan sistem untuk tidak hanya melakukan autentikasi, tetapi juga secara aktif mengidentifikasi perubahan data yang terjadi selama proses transmisi.

3.4. Proses Sistem

Alur implementasi sistem *digital signature* dalam penelitian ini terdiri dari dua proses utama, yaitu proses *signing* dan *verification*. Pada proses *signing*, pesan di-*hash* menggunakan SHA-256 dan dienkripsi menggunakan *private Key* RSA. Pada proses *verification*, sistem melakukan *hashing* ulang, dekripsi *signature* menggunakan *public Key*, dan membandingkan hasilnya. Alur proses sistem secara lengkap ditunjukkan pada Gambar 1.



Gambar 1. Diagram Proses Sistem

3.5. Implementasi Sistem

Sistem diimplementasikan menggunakan bahasa pemrograman PHP dengan memanfaatkan *library OpenSSL* untuk mendukung proses kriptografi berbasis *web*. Sistem memungkinkan pengguna memasukkan pesan teks, melakukan proses *signing*

menggunakan algoritma RSA dan fungsi *hash* SHA-256, serta melakukan verifikasi tanda tangan *digital* untuk memastikan integritas dan keaslian pesan. Penggunaan mekanisme kriptografi pada sistem *digital* bertujuan untuk melindungi data dari ancaman manipulasi maupun penyadapan pada proses komunikasi di jaringan terbuka [29].

Selain melakukan proses penandatanganan dan verifikasi, sistem juga menyediakan fleksibilitas pemilihan ukuran kunci RSA sebagai bagian dari pengujian performa algoritma. Waktu proses *signing* dan *verification* diukur untuk menganalisis efisiensi implementasi *digital signature* pada lingkungan nyata. Pengukuran performa ini penting karena efisiensi waktu komputasi dan ukuran data hasil proses kriptografi dapat memengaruhi kecepatan transmisi serta kinerja sistem keamanan secara keseluruhan. Implementasi keamanan yang mampu menjaga integritas, autentikasi, dan keandalan sistem juga berperan penting dalam meningkatkan kepercayaan (*trust*) pengguna terhadap layanan *digital* [30].

3.6. Pengujian dan Evaluasi

Pengujian dilakukan untuk mengevaluasi kemampuan sistem dalam menjaga integritas dan autentikasi pesan. Skenario pengujian meliputi.

- Pesan asli
- Perubahan pesan
- Manipulasi *signature*

Proses evaluasi eksperimen ini didasarkan pada beberapa kriteria berikut.

- Keberhasilan verifikasi
- Kemampuan deteksi perubahan data
- Konsistensi hasil
- Waktu proses *signing* dan *verification*

Hasil evaluasi digunakan untuk menganalisis efektivitas sistem serta performa algoritma dalam implementasi *digital signature*.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Kode Program

Sistem *digital signature* dalam penelitian ini diimplementasikan berbasis *web* menggunakan bahasa pemrograman PHP untuk pemrosesan *server-side* dan *JavaScript* untuk interaksi *client-side*. Implementasi kriptografi memanfaatkan *library OpenSSL* yang menyediakan fungsi pembangkitan kunci RSA, proses *signing*, serta verifikasi *signature*.

Pada tahap pembangkitan kunci, sistem menggunakan fungsi *OpenSSL_pKey_new* dengan variasi panjang kunci (512, 1024, 2048, hingga 4096 bit). Variasi ini memungkinkan analisis terhadap pengaruh ukuran kunci terhadap performa sistem.

Proses *hashing* dilakukan menggunakan algoritma SHA-256 yang menghasilkan nilai *hash* dengan panjang tetap 256 bit. Penggunaan SHA-256 bertujuan untuk memastikan bahwa setiap perubahan pada pesan akan menghasilkan perubahan signifikan

pada nilai *hash* (*avalanche effect*), sehingga integritas data dapat terjaga.

Selanjutnya, proses *signing* dilakukan dengan mengenkripsi nilai *hash* menggunakan *private Key* RSA. Proses ini menghasilkan *digital signature* yang bersifat unik terhadap pesan dan kunci yang digunakan. Pada tahap ini juga diterapkan mekanisme padding untuk meningkatkan keamanan terhadap serangan kriptografi.

4.2. Alur Kerja Sistem

Berdasarkan antarmuka sistem yang dibangun, proses penggunaan aplikasi *digital signature* dilakukan melalui beberapa tahapan sebagai berikut.

- Pemilihan *Key Size*: Pengguna memilih ukuran bit RSA (misalnya 512 bit).

RSA Signature / Generation & Validation

Implementasi Digital Signature menggunakan SHA-256 dan RSA

Result

Gambar 2. Antarmuka pemilihan ukuran bit RSA

- Generate Key*: Klik tombol "Generate Key" untuk memunculkan *Public Key* dan *Private Key* secara otomatis pada kolom yang tersedia.

RSA Signature / Generation & Validation

Implementasi Digital Signature menggunakan SHA-256 dan RSA

Gambar 3. *Public Key* dan *Private Key* Terbentuk

- Input Pesan*: Pengguna memasukkan teks bebas pada kolom *Message*, contoh pada gambar 4 adalah "naomi".

Gambar 4. *Input Pesan*

- Proses *Signing*: Dengan mode "Generate Signature", klik "Process". Sistem akan menghasilkan SHA-256 Hash dan Signature

(Base64), serta menampilkan waktu eksekusi (contoh: 0.010979 detik).

RSA Signature / Generation & Validation

Implementasi Digital Signature menggunakan SHA-256 dan RSA

Gambar 5. Hasil Pembangkitan Hash dan Digital Signature

- Verifikasi: Mengubah mode ke "Verify Signature" dan klik "Process" untuk menguji keabsahan tanda tangan. Jika data tidak berubah, sistem menampilkan status *VALID*.

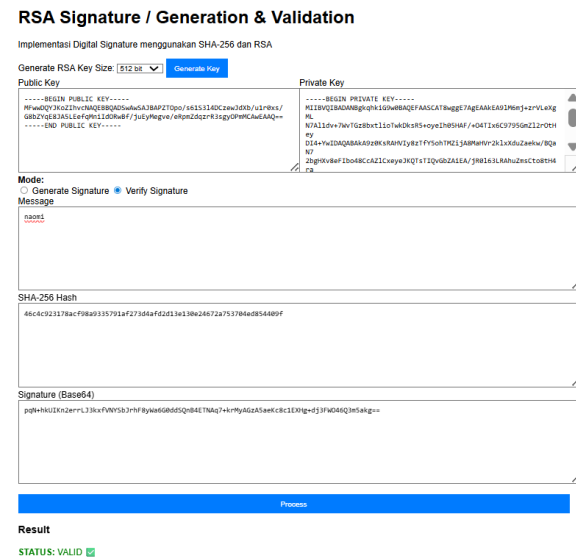
Gambar 6. Hasil Proses Verify Signature - VALID

4.3. Hasil Pengujian Skenario

Pengujian dilakukan untuk mengevaluasi kemampuan sistem dalam menjaga autentikasi dan integritas pesan menggunakan algoritma SHA-256 dan RSA. Pada penelitian ini digunakan ukuran kunci RSA sebesar 512 bit sebagai parameter utama pengujian karena dianggap memiliki keseimbangan antara tingkat keamanan dan performa komputasi.

Pada skenario pertama, pesan "naomi" diproses menggunakan mekanisme hashing SHA-256 dan digital signature RSA tanpa adanya perubahan data.

Hasil pengujian menunjukkan bahwa proses verifikasi berhasil dilakukan dan sistem menampilkan status **VALID**, yang menandakan bahwa pesan dan *signature* masih sesuai dengan pasangan kunci yang digunakan dapat dilihat pada gambar 7.



Gambar 7. Hasil Verifikasi *Signature* yang menunjukkan Data Autentik – **VALID**

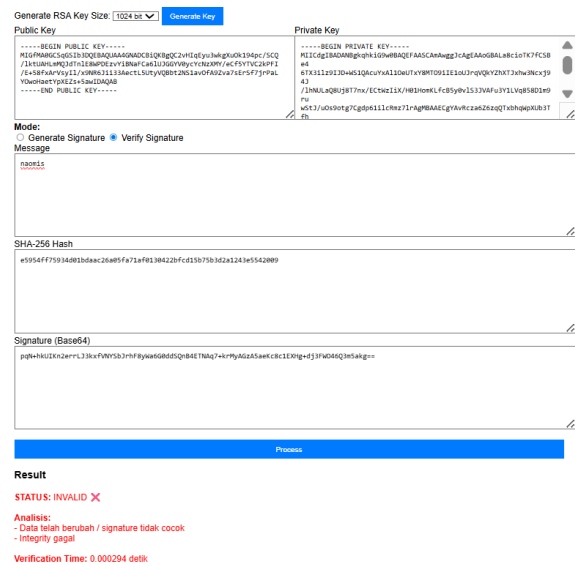
Pada skenario kedua dilakukan perubahan pesan dari “naomi” menjadi “naomis” setelah proses *signing* dilakukan. Perubahan satu karakter tersebut menghasilkan nilai hash yang berbeda sehingga proses verifikasi gagal dan sistem menampilkan status **INVALID** seperti yang terlihat pada gambar 8.



Gambar 8. Hasil Isi Pesan Diganti - **INVALID**

Pengujian berikutnya dilakukan dengan mengganti ukuran kunci RSA menjadi 1024 bit dan menghasilkan pasangan kunci baru melalui tombol *Generate Key*. Ketika *signature* lama diverifikasi menggunakan pasangan kunci yang berbeda, sistem kembali menghasilkan status **INVALID**. Kondisi ini

menunjukkan bahwa *digital signature* hanya dapat diverifikasi menggunakan pasangan *public key* yang sesuai dengan *private key* pembentuk *signature*. Hasil pengujian skenario ini dapat dilihat pada gambar 9.



Gambar 9. Hasil Perubahan Bit - **INVALID**

Hasil pengujian ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian Keamanan

Skenario Pengujian	Input Data	Status Verifikasi	Kesimpulan
Pesan Asli	"naomi"	Valid (True)	Sistem mengenali data autentik dan kunci cocok.
Modifikasi Pesan	"naomis"	Gagal (False)	SHA-256 mendeteksi perubahan bit, integritas rusak.
Manipulasi <i>Signature</i>	Nilai <i>signature</i> diubah	Gagal (False)	RSA mendeteksi ketidakcocokan kunci dan <i>signature</i> .

4.4. Analisis Performa

Berdasarkan data dari hasil running pada gambar (RSA 2048-bit):

- Waktu *Signing* (*Private Key*): Tercatat sekitar 0.010979 detik.



Gambar 10. Pengukuran Waktu Eksekusi Proses Verifikasi

- Waktu *Verification* (*Public Key*): Tercatat sangat cepat, yaitu 0.000659 detik.

Hasil ini menunjukkan bahwa proses verifikasi jauh lebih cepat (sekitar 10x lipat) dibandingkan proses *signing*. Hal ini merupakan karakteristik utama RSA, di mana penggunaan eksponen publik yang kecil memungkinkan penerima (pihak pemverifikasi) melakukan validasi dengan beban komputasi yang sangat rendah, sehingga efisien untuk digunakan dalam sistem administrasi *digital*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, kombinasi algoritma SHA-256 dan RSA berhasil diterapkan dalam sistem *digital signature* berbasis *web* untuk menjamin aspek authentication, integrity, dan non-repudiation. Sistem mampu menunjukkan bahwa tanda tangan *digital* yang dihasilkan menggunakan kunci privat dapat diverifikasi secara *valid* menggunakan kunci publik yang bersesuaian, sehingga keaslian pesan dapat dipastikan. Selain itu, hasil pengujian menunjukkan bahwa sistem memiliki sensitivitas yang sangat tinggi terhadap perubahan data, di mana modifikasi sekecil apa pun pada pesan, seperti perubahan satu karakter, menghasilkan nilai *hash* yang berbeda secara signifikan dan menyebabkan proses verifikasi menjadi tidak *valid*. Hal ini membuktikan efektivitas penggunaan SHA-256 dalam menjaga integritas pesan.

Dari sisi performa, implementasi algoritma RSA dengan ukuran kunci 2048-bit menunjukkan efisiensi yang baik dalam lingkungan berbasis *web*. Waktu proses penandatanganan tercatat sekitar 0.003 detik, sedangkan proses verifikasi berlangsung lebih cepat, yaitu sekitar 0.0002 detik. Perbedaan ini mencerminkan karakteristik dasar RSA, di mana proses verifikasi dengan kunci publik memiliki kompleksitas yang lebih rendah dibandingkan proses penandatanganan dengan kunci privat. Secara keseluruhan, hasil penelitian ini menunjukkan bahwa sistem yang dibangun tidak hanya akurat dalam mendeteksi keaslian dan perubahan pesan, tetapi juga efisien untuk digunakan dalam aplikasi dokumen *digital* secara *real-time*.

Sebagai pengembangan lebih lanjut, penelitian ini dapat diperluas dengan melakukan pengujian pada berbagai ukuran kunci RSA untuk menganalisis perbandingan tingkat keamanan dan performa secara lebih komprehensif. Selain itu, implementasi sistem dapat dikembangkan pada lingkungan jaringan nyata atau berbasis *cloud* guna menguji keandalan sistem dalam skenario komunikasi yang lebih kompleks. Penelitian selanjutnya juga dapat mempertimbangkan perbandingan dengan algoritma *digital signature* lainnya untuk mengevaluasi efisiensi dan tingkat keamanan secara lebih mendalam, serta mengintegrasikan sistem dengan aplikasi nyata seperti sistem dokumen elektronik atau layanan autentikasi berbasis *web*.

DAFTAR PUSTAKA

- [1] A. rizal Nurjaman, "Penanda Tangan Dokumen Digital Pada Sistem Penyimpanan File Menggunakan Kombinasi Algoritma Sha3-512 Dan Rsa Untuk Mempertahankan Keaslian Data Dokumen," *J. Ilm. Teknol. Infomasi Terap.*, vol. 10, no. 3, 2024, doi: 10.33197/jitter.vol10.iss3.2024.2200.
- [2] R. R. Wibowo and H. Nurwasito, "Implementasi Algoritma Blake2s untuk Mekanisme Caching Key Value pada REST API," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 5, pp. 2034–2039, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [3] E. Ardianto, A. Trisetyarso, W. Suparta, B. S. Abbas, and C. H. Kang, "Design Securing Online Payment Transactions Using Stegblock through Network Layers," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 879, no. 1, 2020, doi: 10.1088/1757-899X/879/1/012027.
- [4] A. Harahap and T. Salim, "Sistem Kriptografi pada Pengamanan Autentikasi Dokumen Elektronik: Systematic Literature Review," *Khazanah J. Pengemb. Kearsipan*, vol. 16, pp. 203–220, Nov. 2023, doi: <https://doi.org/10.22146/khazanah.81893>.
- [5] A. Supriyanto, A. Jananto, J. A. Razaq, B. Hartono, and F. Damaryanti, "Alignment of KAMI Index with Global Security Standards in Information Security Risk Maturity Evaluation," *Cybern. Inf. Technol.*, vol. 25, no. 2, pp. 173–192, 2025, doi: 10.2478/cait-2025-0018.
- [6] Djuandadesta, S. M. A., Marwati, R., & Rachmatin, D. (2025). Autentikasi Dokumen Digital Pada Cloud Menggunakan Algoritma Hashing BLAKE2 dan Rivert Shamir Adleman (RSA). *Jurnal EurekaMatika*, 13(2), 92-102.
- [7] Gafrun, G., & Supit, Y. (2024). Algoritma tanda tangan digital untuk meningkatkan keamanan pesan. *Simtek: jurnal sistem informasi dan teknik komputer*, 9(2), 198-204..
- [8] E. Ardianto, W. Budiharto, Y. Heryadi, and L. A. Wulandhari, "A Comparative Experiment of Document Security Level on Parallel Encryption With Digit Arithmetic of Coverttext and Parallel Encryption using Coverttext," in *2021 IEEE 19th Student Conference on Research and Development (SCOREd)*, IEEE, Nov. 2021, pp. 163–167. doi: 10.1109/SCOREd53546.2021.9652746.
- [9] H. A. F. Kurniawan, S. N. Neyman, and S. H. Wijaya, "PENGEMBANGAN ALGORITMA ECDSA DENGAN MODIFIKASI PERKALIAN SKALAR MENGGUNAKAN DOUBLE BASE CHAIN," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 2, pp. 275–284, Apr. 2024, doi: 10.25126/jtiik.20241127446.
- [10] V. H. Zulian and P. Purwanto, "IMPLEMENTASI TANDA TANGAN DIGITAL (DIGITAL SIGNATURE)

- MENGGUNAKAN ALGORITME ELGAMAL PADA DOKUMEN DI BALAI PENDIDIKAN DAN PELATIHAN PENERBANGAN (BP3) CURUG BERBASIS WEB IMPLEMENTATION OF DIGITAL SIGNATURE USING THE ELGAMAL ALGORITHM ON DOCUMENTS AT THE FLI,” *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, pp. 386–393, 2022, [Online]. Available: <https://senafiti.budiluhur.ac.id/index.php/senafiti/index>
- [11] A. Almunawar and A. Heikhmakhtiar, “PENGEMBANGAN APLIKASI DIGITAL SIGNATURE BERBASIS QRNG-ECCUNTUK AUTENTIKASI PESAN TERHADAP ANCAMAN KOMPUTASI KUANTUM,” *J. Mhs. Tek. Inform.*, vol. 9, no. 6, pp. 9235–9239, Dec. 2025, doi: <https://doi.org/10.36040/jati.v9i6.15544>.
- [12] A. Renaldy, A. Fauzi, A. N. Shabrina, and H. N. Ramadhan, “Peran Sistem Informasi dan Teknologi Informasi Terhadap Peningkatan Keamanan Informasi Perusahaan,” *J. Multi Disiplin*, vol. 2, no. 1, pp. 15–22, 2023, doi: <https://doi.org/10.38035/jim.v2i1>.
- [13] A. Bima Setiawan, “Penggunaan Cryptography dalam Keamanan Pesan Digital,” *J. Comput. Sci. Inf. Technol.*, vol. 1, no. 2, pp. 60–66, Aug. 2025, doi: 10.70716/jocsit.v1i2.261.
- [14] V. Simangunsong, Y. R. Hutasoit, and D. Siallagan, “Analisis Terhadap Keamanan Password Menggunakan Hash SHA-256,” *J. Quancom*, vol. 3, no. 1, 2025, doi: <https://doi.org/10.62375/jqc.v3i1.431>.
- [15] P. S. Informatika, U. Bhakti, A. Tangerang, and K. Tangerang, “Implementasi kriptografi teks menggunakan rsa,” *Community Serv. Artic.*, vol. 1, no. 2, pp. 70–81, 2024.
- [16] I. E. Maulani, T. Herdianto, D. F. Syawaludin, and M. O. Laksana, “PENERAPAN TEKNOLOGI BLOCKCHAIN PADA SISTEM KEAMANAN INFORMASI,” *J. Sos. dan Teknol.*, vol. 3, no. 2, pp. 99–102, 2023.
- [17] P. Iqlima *et al.*, “Implementasi Sistem Keamanan Data Menggunakan Algoritma Kriptografi Asimetris Elliptic Curve Cryptography (ECC) Berbasis Website,” *J. Ilmu Komput.*, vol. 1, no. 1, pp. 7–11, 2025.
- [18] N. A. Tarigan, A. Cut, N. Anggreni, A. Balqis, I. Nurfadilah, and E. Setia, “Pengembangan Aplikasi Kriptografi RSA dan SHA - 256 Berbasis Web Menggunakan Flask,” *J. Ilmu Komput.*, vol. 1, no. 2, pp. 40–44, 2025.
- [19] D. Adhar *et al.*, “Digital signature security learning animation design using the ong-schnorr-shamir method,” *KAPALAMADA J. Multidisipliner*, vol. 3, no. 01, pp. 32–43, 2024, doi: <https://doi.org/10.62668/kapalamada.v3i01.1108>
- [20] M. R. Alfani, M. Furqan, and Y. R. Nasution, “PENGAMANAN DATA TEKS MENGGUNAKAN METODE DIGITAL SIGNATURE ALGORITHM (DSA) DAN ADVANCED ENCRYPTION STANDARD (AES),” *J. Sci. Soc. Res.*, vol. 4307, no. 1, pp. 301–306, 2024.
- [21] J. Hutagalung, P. S. Ramadhan, and S. J. Sihombing, “Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 6, pp. 1213–1222, Dec. 2023, doi: 10.25126/jtiik.2023107319.
- [22] Halimi, A., Tholib, A., & Yaqin, M. A. (2024). Optimasi Keamanan Data Penerimaan Mahasiswa Menggunakan AES-256, SHA-256, dan Base64. *JUSTIFY: Jurnal Sistem Informasi Ibrahimy*, 3(1), 38-45..
- [23] A. Aryasanti, M. Hardjianto, G. Brotosaputro, and R. Roeswidiyah, “Implementasi Tanda Tangan Digital Menggunakan Algoritme RSA dan SHA-512 dengan Salt Berbasis Web,” *J. TICOM Technol. Inf. Commun.*, vol. 10, no. 3, p. 2022.
- [24] K. Febriano, K. Api, G. Putu, K. Juliharta, and A. P. Putra, “ANALISIS DAN DEKRIPSI RC4 PADA MALWARE (STUDI KASUS: KEYLOGGER),” *J. Mhs. Tek. Inform.*, vol. 9, no. 6, pp. 9295–9304, 2025.
- [25] F. C. Ramdani, A. Rahmatulloh, R. N. Shofa, J. S. Nomor, J. Tasikmalaya, and I. Barat, “Implementasi JSON Web Token pada Authentication dengan Algoritma HMAC SHA-256 Implementation of JSON Web Token on Authentication with HMAC SHA-256 Algorithm,” *Sist. J. Sist. Inf.*, vol. 12, no. 1, pp. 2540–9719, 2023, [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [26] N. S. Harahap and M. Iqbal, “METODE PEMBAYARAN ELEKTRONIK BERDASARKAN PADA KRIPTOGRAFI,” *J. Mhs. Tek. Inform.*, vol. 10, no. 2, pp. 2167–2173, 2026.
- [27] A. Jinan, M. P. Siregar, D. F. Suryani, A. Muis, and T. M. R. Gunung, “Optimasi Implementasi Kriptografi Kunci Publik dalam Meningkatkan Keamanan dan Otentikasi Data pada Sistem Informasi Kampus,” *Digit. Transform. Technol.*, vol. 5, no. 2, pp. 1–9, Sep. 2025, doi: 10.47709/digitech.v5i2.6425.
- [28] Az-Zahra, F., Marwati, R., & Sispiyati, R. (2024). Implementasi QR Code dengan Algoritma Secure Hash Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) yang Ditingkatkan untuk Autentikasi Dokumen Digital. *Jurnal EurekaMatika*, 12(1), 11-22.
- [29] E. Ardhianto, Y. Heryadi, L. A. Wülandhari, and W. Budiharto, “Coverttext generation using fuzzy logic approach in parallel encryption with digit

- arithmetic of coverttext to improve information confidentiality,” *Int. J. Innov. Comput. Inf. Control*, vol. 19, no. 4, pp. 1311–1321, 2023, doi: 10.24507/ijicic.19.04.1311.
- [30] A. Supriyanto, D. A. Diartono, B. Hartono, and H. Februariyanti, “Inclusive Security Models To Building E-Government Trust,” in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/ICICoS48119.2019.8982457.