JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Penerapan Tanda Tangan Digital Mengunakan Algoritma SHA - 256 Dan RSA Untuk Pengamanan Surat Perintah Perjalanan Dinas Di Desa Namo Rambe

Rada Petrata Barus[1], Mhd. Gilang Suryanata[2], Deski Helsa Pane[3], Afdal Al Hafiz[4]
[1,2,3,4] Sistem Informasi, STMIK Triguna Dharma
Email: [1]radabarus12@gmail.com, [2]suryanatagilang@gmail.com, [3]deskihelsa@gmail.com, afdal.alhafiz@trigunadharma.ac.id[4]
Email Penulis Korespondensi: radabarus12@gmail.com

Abstrak
Kemajuan teknologi jaringan komputer saat ini memungkinkan pertukaran informasi yang lebih cepat dan efisien, namun juga menghadirkan tantangan besar dalam menjaga keamanan dan kerahasiaan data. Di Kantor Desa Namorambe, proses pembuatan Surat Perintah Perjalanan Dinas (SPPD) masih dilakukan secara manual, yang rentan terhadap pemalsuan dan inefisiensi. Masalah ini dapat berdampak negatif pada keaslian dokumen dan keuangan pemerintahan desa. Penelitian ini bertujuan untuk mengembangkan sistem pengamanan SPPD dengan menerapkan tanda tangan digital menggunakan algoritma Secure Hash Algorithm 256 (SHA-256) dan Rivest-Shamir-Adleman (RSA). Sistem ini juga menggunakan Quick Response Code (QR Code) untuk memverifikasi otentikasi dan integritas dokumen. Hasil penelitian menunjukkan bahwa penerapan tanda tangan digital berbasis algoritma SHA-256 dan RSA berhasil meningkatkan keamanan dan efisiensi dalam pembuatan dan verifikasi SPPD di Desa Namorambe. Dokumen yang dihasilkan mampu memenuhi syarat autentikasi, integritas, dan keabsahan, sehingga dapat dipercaya oleh penerima dan mencegah terjadinya pemalsuan.

Kata Kunci: Keamanan Informasi, Tanda Tangan Digital, SHA-256, RSA, SPPD, QR Code, Kriptografi

Abstract
The advancement of computer network technology today enables faster and more efficient information exchange, but also presents significant challenges in maintaining data security and confidentiality. At the Namorambe Village Office, the process of issuing Travel Order Letters (SPPD) is still done manually, making it vulnerable to forgery and inefficiency. This issue can negatively impact the authenticity of documents and the financial management of the village administration. This study aims to develop a security system for SPPD by implementing digital signatures using the Secure Hash Algorithm 256 (SHA-256) and Rivest-Shamir-Adleman (RSA) algorithms. The system also utilizes Quick Response Code (QR Code) to verify the authentication and integrity of the documents. The results of the study indicate that the implementation of digital signatures based on the SHA-256 and RSA algorithms successfully enhances the security and efficiency of the creation and verification of SPPD in Namorambe Village. The generated documents meet the requirements of authentication, integrity, and validity, ensuring their reliability and preventing forgery.

Keywords*: Information Security, Digital Signature, SHA-256, RSA, SPPD, QR Code, Cryptography*

1. PENDAHULUAN

Teknologi jaringan komputer pada saat ini dapat menghubungkan antara komputer satu dengan komputer lainnya, untuk saling bertukar informasi. Dengan adanya perkembangan teknologi digital hal yang perlu diperhatikan adalah keamanan dan perlindungan data. Pada saat ini terdapat banyak cara untuk mengirim pesan, menerima pesan, dan setiap orang mendapatkan kemudahan untuk berkomunikasi. Dari segala kecanggihan yang ada terdapat beberapa hal yang sjarang diperhatikan oleh pemilik informasi, yaitu keamanan informasi. Dengan adanya keamanan pada informasi pihak yang tidak berkepentingan tidak dapat mengakses informasi tersebut.

Banyak cara untuk mengamankan informasi salah satunya dengan menggunakan teknik *Kriptografi*. *Kriptografi*, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita, dalam *Kriptografi* data yang dianggap rahasia akan disamarkan dengan sedemikian rupa sehingga kalau pun data itu bisa didapatkan maka tidak akan bisa dimengerti oleh pihak yang tidak berhak. Oleh karena itu diciptakan dan terus dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan data supaya data yang di kirim tidak dapat terbaca oleh pihak lain atau yang dikenal dengan istilah *enkripsi*.

Setiap organisasi pasti memiliki sistem informasinya sendiri. Begitu juga pada Kantor Desa Namorambe. Selama ini, pada Kantor Desa Namorambe, SPPD dikeluarkan setiap kali ada perjalanan dinas. Surat ini penting karena diperlukan untuk dokumentasi dan pencatatan dalam urusan kearsipan di Desa Namorambe baik di Kecamatan maupun tingkat Provinsi. Proses pembuatan surat perintah perjalanan dinas tersebut masih dilakukan secara sederhana sehingga tidak efisien dari segi waktu dan dapat merugikan Struktur Pemerintahan Desa Namorambe jika SPPD di palsukan. Sehingga berdampak besar bagi Pemerintahan Desa Namorambe diantara lain dapat merugikan dalam finansial Pemerintahan Desa Namorambe. Oleh karena itu, Desa Namorambe membutuhkan sebuah sistem terkomputerisasi dalam proses pembuatan SPPD menjadi lebih efektif. Untuk mengatasi hal itu, maka dirasa perlu untuk membuat aplikasi khusus dalam pengembangan sistempembuatan surat perjalanan dinas secara terkomputerisasi dengan mengunakan tanda tangan digital berbasis *QR Code*. Sehingga dapat memudahkan instansi untuk membuat surat perjalanan dinas dan keamanan pengelolaan data perjalanan dinas.

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Keamanan yang perlu dilindungi tersebut berupa tanda tangan kepala desa dan stempel dari desa, khususnya di Desa Namorambe, yaitu dengan menggunakan *Quick Response Code* atausering disebut *QR Code*. *QR Code* adalah kode matriks atau barcode dua dimensiyang dapat diuraikan dengan cepat dan tepat sehingga mampu mempercepat dan mempermudah proses transaksi dengan cara membaca beberapa komponen pada kotak kode.

Surat Perintah Perjalanan Dinas (SPPD) untuk menjaga keaslian dari SPPD tersebut, maka dikembangkanlah tanda tangan digital (*tedigital signature*). Tanda tangan digital merupakan mekanisme otentikasi yang memungkinkan pembuat pesan menambahkan sebuah kode yang bertidak sebagai tanda tangannya. *Digital signature* dapat digunakan untuk melakukan suatu pembuktian secara matematis bahwa pesan tidak mengalami perubahan secara ilegal, sehingga dapat digunakan untuk memverifikasi.

Dengan adanya tanda tangan digital mengunakan *QR Code* penerima pesan akan percaya bahwa pesan yang dikirimkan masih otentik dari pengirim aslinya. Dari aspek keamanan informasi diatas merupakan layanan yang disediakan oleh *kriptografi*. Pada masalah ini algoritma yang digunakan adalah algoritma *Secure hash Algorthm*-256 dan *Rivest Shamir Adleman* [1].

Adapun penelitian terkait sebelumnya membahas tentang algoritma RSA dan *digital signature* yaitu, Pengamanan Dokumen Menggunakan Metode RSA Berbasis Web [2], Tanda Tangan Digital Menggunakan Algoritma acak dan RSA [3]. Penerapan *Digital Signature* dan Kriptografi Pada Otentikasi Sertifikat Tanah Digital, Penerapan Algoritma RSA dan DES Pada Pengamanan File Teks, Model Keamanan Informasi Berbasis *Digital Signature* [4] Dengan Algoritma RSA *Implementasi Kriptografi Algoritma* RSA Dengan *Playfair Cipher* Pada Pesan Teks Berbasis Android [5], *Implementasi Algoritma Kriptografi* RSA Pada Tanda Tangan Digital, dan *Implementasi* Tanda Tangan Digital Menggunakan RSA dan SHA-512 Pada Proses Legalisasi Ijazah.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Dalam teknik pengumpulan data dan informasi, ada 2 teknik yang dilakukan dalam penelitian ini yaitu:

1. Observasi

   Observasi adalah teknik pengumpulan informasi atau data dengan melakukan tinjauan langsung ketempat studi kasus dimana akan dilakukan penelitian. Oleh sebab itu, peneliti melakukan tinjauan langsung ke Kantor Desa Namorambe untuk mengetahui masalah yang terjadi dan apa solusi untuk kendala yang dihadapi mengenai tanda tangan digital SKL.

2. Wawancara

   Wawancara yang digunakan ialah wawancara terstruktur. Artinya pewawancara sudah menyusun terlebih dahulu sejumlah pertanyaan yang akan disampaikan kepada narasumber. Narasumber merupakan informan yang memiliki keterlibatan langsung dengan kegiatan yang ada di Desa Namorambe yaitu Bapak Jon Filter Sembiring sebagai kepala Desa Namorambe. Berikut ini merupakan tampilan SPPD, yang dikeliarkan oleh Desa Namorambe.

Gambar 1 Tampilan SPPD Desa Namorambe

Data yang diambil dari SPPD dan akan diubah ke dalam bentuk tanda tangan digital.

Tabel 1 Data Awal

1	Nomor Surat	094/214/2022

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

2.2 Kritptografi

Kata cryptography berasal dari bahasa Yunani "crypto" artinya tersembunyi atau rahasia (hidden atau secret) dan "graphia" artinya tulisan (writing). Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (message). Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi [6].

Kriptografi memiliki 4 komponen utama yaitu: Plaintext, yaitu pesan yang dapat dibaca; Ciphertext, yaitu pesan sandi/ pesan acak yang tidak bisa dibaca: Key, yaitu kunci untuk melakukan teknik kriptografi, Algoritma, yaitu metode untuk melakukan enkripsi dan dekripsi [7].

Algoritma kriptografi dikelompokkan ke dalam dua jenis yaitu algoritma kriptografi klasik dan algoritma kriptografi modern. Dalam pengoperasiannya, algoritma kriptografi klasik bekerja menggunakan mode karakter sedangkan algoritma kriptografi modern bekerja menggunakan mode bit [8]. Kriptografi memiliki dua konsep utama, yaitu enkripsi (encryption) dan dekripsi (decryption). Enkripsi adalah proses penyandian plainteks menjadi cipherteks, sedangkan dekripsi adalah proses mengembalikan cipherteks menjadi plainteks semula. Enkripsi dan dekripsi membutuhkan kunci sebagai parameter yang digunakan untuk transformasi.

2.3 Secure hash Algorthm-256 (SHA-256)

SHA-256 adalah sebuah kriptografi fungsi *hash* yang dirancang oleh *National Security Agency* (NSA) dan dipublikasikan oleh *National Institute of Standart and Technology* (NIST) sebagai sebuah *Federal Information Processing Standart* (FIPS) oleh U.S. Fungsi utama SHA-256 dapat dilihat pada Gambar 2.2 Fungsi utama SHA-256 menerima masukan berupa data atau pesan *M* dengan panjang sembarang, lalu akan menghasilkan nilai *hash h(M)*. SHA-256 menggunakan enam logika yang merupakan kombinasi dasar seperti AND, OR, XOR, pergeseran bit kekanan (*shift right*), dan rotasi bit kekanan (*rotate right*) [9]. Algoritma ini mengubah sebuah *message schedule* yang terdiri dari 64 *element* 32-bit *word*, delapan buah variabel 32-bit, dan variabel penyimpan nilai *hash* 8 buah *word* 32-bit. Hasil akhir dari algoritma ini adalah sebuah *message digest* sepanjang 256-bit. Algoritma SHA-256 memiliki beberapa langkah pengerjaan yaitu sebagai berikut :

1. Ubah pesan ke bentuk biner
   Pesan yang akan digunakan untuk di dapat *massage digest* nya diubah terlebih dahulu ke dalam bentuk biner.
2. Tambahkan bit *Padding*
   Pesan diisi sehingga panjangnya kongruen dengan 448, modulus 512. *Padding* 1 bit ditambahkan di akhir pesan, diikuti oleh banyaknya nol yang diperlukan sehingga panjang bit sama dengan 448 modulus 512.
3. *Parsing* pesan
   Pesan yang telah di *padding* dibagi menjadi *N* buah blok 512 bit : $M_{(1)}$,$M_{(2)}$,..$M_{(n)}$, dengan menambahkan 16 blok 64 bit setiap blok berukuran 32 bit.
4. Inisialisasi Nilai *Hash*
   Nilai *hash* awal $H_{(0)}$ terdiri dari delapan kata 32 bit, dalam bentuk heksadesimal, yaitu sebagai berikut :
   a = $H_0^{(0)}$ = 6A09E667
   b = $H_1^{(0)}$ = BB67AE85
   c = $H_2^{(0)}$ = 3C6EF372
   d = $H_3^{(0)}$ = A54FF53A
   e = $H_4^{(0)}$ = 510E527F
   f = $H_5^{(0)}$ = 9B056887
   g = $H_6^{(0)}$ = 1F83D9AB
   h = $H_7^{(0)}$ = 5BE0CD19
5. Penjadwalan pesan (*massage Schedulle*)
   Mengubah setiap blok pesan menjadi bilangan heksadesimal yang diberi label W_0, W_1, W_2,…W_{63} dengan ketentuan sebagai berikut :

$$Wt=\begin{cases} Mt(t) & 0 \le t \le 15 \\ \sigma 1\,(256)(Wi-2) + Wi-7 + \sigma 0\,(256)(Wi-15 + Wi-16) & 16 \le t \le 63 \end{cases} \quad \text{...[2.1]}$$

   Dimana :
   σ1(256) (Wi-2) = ((Wi-2)ROTR 17) ⊕ ((Wi-2) ROTR 19 ⊕ ((Wi-2)SHR 10) ……[2.2]
   σ0(256) (Wi-15) = ((Wi-15)ROTR 7) ⊕ ((Wi-15) ROTR 18 ⊕ ((Wi-15)SHR 3) ……..[2.3]
   Keterangan :
   Wt = Blok pesan yang baru
   Mt = Blok pesan yang lama
   Wi-2 = Blok pesan dari W ke i-2
   Wi-15 = Blok pesan dari W ke i-15
   ROTR = Rotate Right
   SHR = Shift Right
   ⊕ = Operator *XOR*

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

6. Inisialisasi Variabel Kerja dan Nilai Konstanta
   Melakukan inisialisasi variabel kerja *a, b, c, d, e, f, g,* dan *h* dimana setiap variabel diambil dari inisial *hash value a* = $H_0^{(0)}$, *b* = $H_1^{(0)}$, *c* = $H_2^{(0)}$, *d* = $H_3^{(0)}$, *e* = $H_4^{(0)}$, *f* = $H_5^{(0)}$, *g* = $H_6^{(0)}$, *h* = $H_7^{(0)}$.

Tabel 2 Nilai Konstanta SHA-256

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 Nilai *Hash*
   Menghitung komputasi nilai *hash* dari a sampai nilai h sebanyak 64 kali putaran dengan persamaan berikut :

   $T_1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_t^{(256)} + Wt$ ……..[2.4]

   $T_2 = \sum_0^{(256)}(a) + Maj(a,b,c)$ ……..[2.5]

   h = g
   g = f
   f = e
   e = d + T_1
   d = c
   c = b
   b = a
   a = T_1 + T_2

   Dimana :

   $\sum_1^{(256)}(e)$ = (e *ROTR* 6) ⊕ (e *ROTR* 11) ⊕ (e *ROTR* 25)……..[2.6]

   $\sum_0^{(256)}(a)$ = (e *ROTR* 2) ⊕ (e *ROTR* 13) ⊕ (e *ROTR* 22) ……..[2.7]

   Ch (e, f, g) = (e ^ f) ⊕ (~e ^ g) ……..[2.8]

   Maj (a, b, c) = (a ^ b) ⊕ (a ^ c) ⊕ (b ^ c) ……..[2.9]

   Keterangan :

   a,b,c,d,e,f,g,h = Variabel yang berisi blok – blok decimal
   $K_t^{(256)}$ = Konstanta SHA-256
   W_t = Nilai W pada massage Schadule
   ROTR = *Rotate Right*
   ⊕ = fungsi *XOR*
   ^ = fungsi *AND*
   ~ = Negasi

8. Menjumlahkan hasil akhir *a, b, c, d, e, f, g, h* dengan inisial *hash value* H(i).
   $H_0^{(i)} = a + H_0^{(i)}$
   $H_1^{(i)} = b + H_1^{(i)}$
   $H_2^{(i)} = c + H_2^{(i)}$
   $H_3^{(i)} = d + H_3^{(i)}$
   $H_4^{(i)} = e + H_4^{(i)}$
   $H_5^{(i)} = f + H_5^{(i)}$
   $H_6^{(i)} = g + H_6^{(i)}$
   $H_7^{(i)} = h + H_7^{(i)}$
9. Hasil
   Setelah mengulangi langkah 1 hingga 4 sebanyak N kali, fungsi *hash* yang dihasilkan adalah sebagai berikut :
   H0(N) ‖ H1(N) ‖ H2(N) ‖ H3(N) ‖ H4(N) ‖ H5(N) ‖ H6(N) ‖ H7(N)

2.4 Rivest Shamir Adleman (RSA)

RSA ditemukan oleh tiga orang yang kemudian disingkat menjadi RSA. Ketiga penemu itu adalah Ron Rivest, Adi Shamir, dan Leonard Adleman. RSA termasuk algoritma asimetri, yang berarti memiliki dua kunci, yaitu kunci publik dan kunci privat [10].

RSA menjadi sistem kriptografi kunci publik yang terpopuler karena merupakan sistem pertama yang sekaligus dapat digunakan untuk *key distribution*, *confidentiality* dan *digital signature* [11]. Hal pertama yang dilakukan algoritma pembangkit kunci adalah membangkitkan 2 bilangan prima besar [12]. Pembangkitan bilangan prima besar menggunakan

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

algoritma pengujian bilangan prima misalnya algoritma Miller Rabin. Untuk membangkitkan kunci RSA terdapat beberapa parameter yaitu p. 4. no (n), e, d. K publik, dan K privat. Keterangan dari masing-masing parameter dapatdilihat pada Tabel

Tabel 3 Parameter Pembangkit Kunci RSA

PARAMETER	KETERANGAN
p	Bilangan prima
q	Bilangan prima
n	Merupakan hasil dari $p \times q$
ø *(n)*	Merupakan hasil dari $(p-1) \times (q-1)$
e	Dengan ketentuan, gcd (ø (n), e) = 1 gcd = *greatest common divisor*
d	e^{-1} *(mod ø (n))* Menggunakan algoritma *extended euclid*
K publik	*(e,n)*
K privat	*d*

Contoh pembangkitan kunci publik dan kunci privat sistem kriptografi RSA:
1. Pilih bilangan prima p dan q, misalnya p = 17 dan q = 31
2. Cari $n = p \times q = 17 \times 31 = 527$
3. Hitung Ø $(n) = (p-1)(g-1) = (17-1) \times (31-1)$ -16 – 30 - 480
4. Misalnya $e = 77$, nilai 77 dipilih karena memenuhi syarat gdc (480,77) = 1
5. Didapat $d = e^{-1} = 77^{-1} \bmod 480$, dengan menggunakan algoritma *extended euclid*
   $d = 293$
   1. Maka Kpublik = $(e,n) = (77,527)$ dan $Kprivat = d = 293$

Setelah kunci publik Kpublik dibangkitkan oleh pendekripsi, maka sembarang orang dapat menggunakan kunci publik tersebut. Algoritma enkripsi RSA menggunakan fungsi eksponensial dalam modular n, seperti yang dijelaskan pada Tabel 4.

Tabel 4 Algoritma Enkripsi RSA

Algortima Enkripsi RSA	
Input	*Kpublik = (e,n)*
Output	$C = P^e \bmod n$ Menggunakan algoritma *square and multipy*

Contoh enkripsi RSA, menyambung dari contoh pembangkitan kunci publik dan kunci privat sistem kriptografi RSA sebelumnya:
1. Misal diberikan teks asli P-51
2. Karena kunci publik *Kpublik* = (77,527) berarti $e = 77$ dan n = 527
3. Maka teks sandi C-P mod n=51" mod 527=493

Algoritma dekripsi RSA merupakan fungsi eksponensial modular n dengan menggunakan kunci privat, seperti yang dijelaskanpada Tabel 6.

Tabel 5 Algoritma Deskripsi RSA

Algortima Dekripsi RSA	
Input	*Kprivat = d* *Kpublik = (e,n)*
Output	$P = C^d \bmod n$

Contoh dekripsi RSA, menyambung dari contoh pembangkitan kunci publik dan kunci privat sistem kriptografi RSA dan contoh enkripsi sebelumnya:
1. Telah diketahui kunci privat *Kprivat* = 293 berarti *d*=293
2. Juga telah diketahui $n = 527$ dan teks sandi C = 493
3. Maka $P = C^d \bmod n = 493^{293} \bmod 527 = 51$
4. Jadi teks asli P = 51 bernilai sama dengan teks asli pada contoh enkripsi

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

3. HASIL DAN PEMBAHASAN

3.1 Penerapan Algoritma

Algoritma Sistem adalah langkah-langkah untuk menyelesaikan masalah dalam perancangan sistem penerapan tanda tangan digital mengunakan algoritma SHA - 256 dan RSA untuk pengamanan surat perintah perjalanan dinas di desa Namorambe. Adapun proses atau tahapan pada algoritma sha-256 adalah:

a. Menentukan pesan (M) yang diambil dari data SPPD. Data yang digunakan adalah nomor surat (094/214/2022). M = 094/214/2022. Ubah pesan (M) menjadi biner sehingga menjadi bilangan biner. Sehingga menjadi :

0	= 00110000	2	= 00110010	2	= 00110010
9	= 00111001	1	= 00110001	0	= 00110000
4	= 00110100	4	= 00110100	2	= 00110010
/	= 00101111	/	= 00101111	2	= 00110010

Maka M = 00110000 00111001 00110100 00101111 00110010 00110001
00110100 00101111 00110010 00110000 00110010 00110010

b. *Padding*, adalah penambahan bit pengganjal sehingga total panjangnya 512 bit.
M = 00110000 00111001 00110100 00101111 00110010 00110001
00110100 00101111 00110010 00110000 00110010 00110010
Maka panjang pesan, ℓ = 96 bit.

Padding dilakukan dengan cara menambahkan bit '1' dan sisanya adalah bit '0' sejumlah k. Dimana k :

$$\ell + \mathbf{1} + \mathbf{k} = \mathbf{448\ mod\ 512}$$

k = ℓ + 1 = 488 mod 512
= 96 + 1 = 488 mod 512
= 97 = 448 mod 512
= 448 – 97 = **351.** Maka banyak bit '0' yang ditambahkan sebanyak 351 bit

Setelah itu tambahkan jumlah panjang pesan pada akhir pesan yang di *padding*.
ℓ = 96 = 01100000
Hasil pesan yang di *padding* [$M^{(0)}$]

Tabel 6 Hasil *padding* pesan

00110000	00111001	00110100	00101111	00110010	00110001	00110100	00101111
00110010	00110010	00110010	00110010	10000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	01100000

c. *Parsing* pesan, adalah membagi setiap blok 512 bit menjadi 16 buah *word* 32 bit.

Tabel 7 Hasil *Parsing* pesan

$M_0^{(0)}$	0011	0000	0011	1001	0011	0100	0011	0100
$M_1^{(0)}$	0011	0010	0011	0001	0011	0100	0011	0100
$M_2^{(0)}$	0010	1111	0011	0010	0011	0010	0011	0010
$M_3^{(0)}$	1000	0000	0000	0000	0000	0000	0000	0000
$M_4^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_5^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_6^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_7^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_8^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_9^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_{10}^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_{11}^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_{12}^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_{13}^{(0)}$	0000	0000	0000	0000	0000	0000	0000	0000
$M_{14}^{(0)}$	0000	0000	0000	0000	00000	0000	0000	0000
$M_{16}^{(0)}$	0000	0000	0000	0000	0000	0000	0110	0000

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

d. *Massage Schedule*. Langkah ini diawali dengan mengubah setiap blok pesan menjadi bilangan *hexadecimal* dengan ketentuan sebagai berikut:

$$W_t = \begin{cases} Mt(t) & 0 \leq t \leq 15 \\ \sigma 1\,(256)(Wi-2) + Wi - 7 + \sigma 0\,(256)(Wi-15 + Wi-16) & 16 \leq t \leq 63 \end{cases}$$

Tabel 8 Hasil *Massage Schedule*

W_0	3039342F	W_{16}	00000000	W_{32}	00000000	W_{48}	00000000
W_1	3231342F	W_{17}	00000000	W_{33}	00000000	W_{49}	00000000
W_2	32323232	W_{18}	00000000	W_{34}	00000000	W_{50}	00000000
W_3	80000000	W_{19}	00000000	W_{35}	00000000	W_{51}	00000000
W_4	00000000	W_{20}	00000000	W_{36}	00000000	W_{52}	00000000
W_5	00000000	W_{21}	00000000	W_{37}	00000000	W_{53}	00000000
W_6	00000000	W_{22}	00000000	W_{38}	00000000	W_{54}	00000000
W_7	00000000	W_{23}	00000000	W_{39}	00000000	W_{55}	00000000
W_8	00000000	W_{24}	00000000	W_{40}	00000000	W_{56}	00000000
W_9	00000000	W_{25}	00000000	W_{41}	00000000	W_{57}	00000000
W_{10}	00000000	W_{26}	00000000	W_{42}	00000000	W_{58}	00000000
W_{11}	00000000	W_{27}	00000000	W_{43}	00000000	W_{59}	00000000
W_{12}	00000000	W_{28}	00000000	W_{44}	00000000	W_{60}	00000000
W_{13}	00000000	W_{29}	00000000	W_{45}	00000000	W_{61}	00000000
W_{14}	00000000	W_{30}	00000000	W_{46}	00000000	W_{62}	00000000
W_{15}	00000060	W_{31}	00000000	W_{47}	00000000	W_{63}	00000000

e. Inisialisasi Variabel dan Konstanta

Variabel awal pada fungsi *hash* SHA-256 adalah sebagai berikut :

a = $H_0^{(0)}$ = 6A09E667
b = $H_1^{(0)}$ = BB67AE85
c = $H_2^{(0)}$ = 3C6EF372
d = $H_3^{(0)}$ = A54FF53A
e = $H_4^{(0)}$ = 510E527F
f = $H_5^{(0)}$ = 9B056887
g = $H_6^{(0)}$ = 1F83D9AB
h = $H_7^{(0)}$ = 5BE0CD19

Nilai konstanta pada fungsi *hash* SHA-256 adalah sebagai berikut :

Tabel 9 Nilai Konstanta fungsi *hash* SHA-256

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

f. Komputasi *hash*. Melakukan proses komputasi untuk t = 0, sampai t = 63 sesuai dengan fungsi SHA-256. Dimana fungsi SHA-256 adalah sebagai berikut.

For t = 0 to t = 63 :
{
$T_1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_t^{(256)} + Wt$
$T_2 = \sum_0^{(256)}(a) + Maj(a,b,c)$
h = g
g = f
f = e
e = d + T_1
d = c
c = b
b = a
a = T_1 + T_2

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Keterangan :

a,b,c,d,e,f,g,h	= Variabel yang berisi blok – blok decimal
$\sum_1^{(256)}(e)$	= (e *ROTR* 6) ⊕ (e *ROTR* 11) ⊕ (e *ROTR* 25)
$\sum_0^{(256)}(a)$	= (e *ROTR* 2) ⊕ (e *ROTR* 13) ⊕ (e *ROTR* 22)
$K_t^{(256)}$	= Konstanta SHA-256
W_t	= Nilai W pada massage Schadule
Ch (e, f, g)	= (e ^ f) ⊕ (~e ^ g)
Maj (a, b, c)	= (a ^ b) ⊕ (a ^ c) ⊕ (b ^ c)
ROTR	= *Rotate Right*
⊕	= fungsi *XOR*
^	= fungsi *AND*

Tabel 10 Proses Komputasi Fungsi *HASH* SHA-256

	A	B	C	D	E	F	G	H
Init	6A09E667	BB67AE85	3C6EF372	A54FF53A	510E527F	9B056887	1F83D9AB	5BE0CD19
t=0	2C41BC7C	6A09E667	BB67AE85	3C6EF372	A54FF53A	510E527F	9B056887	1F83D9AB
t=1	2E39BC7C	2C41BC7C	6A09E667	BB67AE85	3C6EF372	A54FF53A	510E527F	9B056887
t=2	2E3ABA7F	2E39BC7C	2C41BC7C	6A09E667	BB67AE85	3C6EF372	3C6EF372	510E527F
t=3	7C08884D	2E3ABA7F	2E39BC7C	2C41BC7C	6A09E667	BB67AE85	BB67AE85	3C6EF372

Selanjutnya dilakukan lagi pencarian nilai a untuk putaran ke empat dan seterusnya hingga putaran 64 (t = 63). Setelah di dapat ke 64 putaran dari kompulasi fungsi *hash*, kemudian dilakukan penjumlahan hasil putaran ke 64 (t = 63) dengan variabel awal *hash* SHA-256. Maka hasil yang didapat adalah sebagai berikut :

$H_0^{(0)}$ = 0AA76655 + 6A09E667 = 74B14CBC
$H_1^{(0)}$ = 5806ED0F + BB67AE85 = 136E9B94
$H_2^{(0)}$ = 5C529E14 + 3C6EF372 = 98C19186
$H_3^{(0)}$ = 742475A6 + A54FF53A = 1F746AE0
$H_4^{(0)}$ = 61D222F0 + 510E527F = B2EO756F
$H_5^{(0)}$ = 5BE71946 + 9B056887 = F6EC81CD
$H_6^{(0)}$ = FF7F9BE7 + 1F83D9AB = 1F037592
$H_7^{(0)}$ = 956297BC + 5BE0CD19 = F14364D5

g. Penggabungan H0 – H7
$H_0 \| H_1 \| H_2 \| H_3 \| H_4 \| H_5 \| H_6 \| H_7$
74B14CBC ‖ 136E9B94 ‖ 98C19186 ‖ 1F746AE0 ‖ B2EO756F ‖F6EC81CD ‖ 1F037592 ‖ F14364D5

h. Nilai *Hash*
74B14CBC136E9B9498C191861F746AE0B2EO756FF6EC81CD1F037592F143B4D5.

1. Bangkitkan 2 bilangan prima sembarang p dan q, misalnya p = 7 dan q = 13.
2. Cari n = p * q
   n = 7 * 13
   n = 91
3. Cari φ (n) = (p – q) * (q – 1)
   φ (n) = (7 – 1) * (13 – 1)
   φ (n) = 72
4. Pilih e yang nilai nya relative prima terhadap φ (n), yaitu 5. 5 relatif prima terhadap 72.
5. Tentukan kunci privat d dengan persamaan d = (1 + a* (n))/e, dengan a adalah bilangan bulat yang dapat memenuhi. Sehingga untuk proses mencari nilai d sebagai berikut :

a = 0 => d = 1/5 (tidak memenuhi)
a = 0 => d = 75/5 (tidak memenuhi)
a = 0 => d = 145/5 = 29 (memenuhi)
Sehinggga didapatkan kunci public (n = 91, e = 5) dan kunci privat (n= 91, =29)

Setelah e-dokumen telah diubah menjadi *massage digest* maka proses selanjutnya adalah mengenkripsi *massage digest* mengunakan algoritma RSA dan kunci privat yang telah dibangkitkan sebagai contoh pesan yang akan dienkripsi adalah hasil *hash* pada contoh *hash* diatas yaitu **“74B14CBC136E9B9498C191861F746AE0B2EO756FF6EC81CD1F037592F143B4D5”.**

1. Pertama kita akan mengubah setiap karakter menjadi desimal melalui tabel ASCII. “55 52 66 49 52 67 66 67 49 51 54 69 57 66 57 52 57 56 67 49 57 49 56 54 49 70 55 52 54 65 69 48 66 50 69 48 55 53 54 70 70 54 69 67 56 49 67 68 49 70 48 51 55 53 57 50 70 49 52 51 54 52 68 53”.
2. Selanjutnya ubah ke *chipertext* dengan rumus $C = M^e \bmod n$

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Tabel 11 Hasil *Chipertext*

55^5 mod 91 = 48	52^5 mod 91 = 26	66^5 mod 91 = 40	49^5 mod 91 = 56
52^5 mod 91 = 26	67^5 mod 91 = 58	66^5 mod 91 = 40	67^5 mod 91 = 58
49^5 mod 91 = 56	51^5 mod 91 = 25	54^5 mod 91 = 45	69^5 mod 91 = 62
57^5 mod 91 = 57	66^5 mod 91 = 40	57^5 mod 91 = 57	52^5 mod 91 = 26
57^5 mod 91 = 57	56^5 mod 91 = 49	67^5 mod 91 = 58	49^5 mod 91 = 56
57^5 mod 91 = 57	49^5 mod 91 = 56	56^5 mod 91 = 49	54^5 mod 91 = 45
49^5 mod 91 = 56	70^5 mod 91 = 70	55^5 mod 91 = 48	52^5 mod 91 = 26
54^5 mod 91 = 45	65^5 mod 91 = 39	69^5 mod 91 = 62	48^5 mod 91 = 55
66^5 mod 91 = 40	50^5 mod 91 = 85	69^5 mod 91 = 62	48^5 mod 91 = 55
55^5 mod 91 = 48	53^5 mod 91 = 79	54^5 mod 91 = 45	70^5 mod 91 = 70
70^5 mod 91 = 70	54^5 mod 91 = 45	69^5 mod 91 = 62	67^5 mod 91 = 58
56^5 mod 91 = 49	49^5 mod 91 = 56	67^5 mod 91 = 58	68^5 mod 91 = 87
49^5 mod 91 = 56	70^5 mod 91 = 70	48^5 mod 91 = 55	51^5 mod 91 = 25
55^5 mod 91 = 48	53^5 mod 91 = 79	57^5 mod 91 = 57	50^5 mod 91 = 85
70^5 mod 91 = 70	49^5 mod 91 = 56	52^5 mod 91 = 26	51^5 mod 91 = 25
54^5 mod 91 = 45	52^5 mod 91 = 26	68^5 mod 91 = 87	53^5 mod 91 = 79

Sehingga diperoleh nilai *chipertext* nya yaitu :
"48 26 40 56 26 58 40 58 56 25 45 62 57 40 57 26 57 49 58 56 57 56 49 45 56 70 48 26 45 39 62 55 40 85 62 55 48 79 45 70 70 45 62 58 49 56 58 87 56 70 55 25 48 79 57 85 70 56 26 25 45 26 87 79".

Proses verifikasi dokumen untuk membuktikan apakah dokumen sah atau tidak dimulai dengan merubah tanda tangan digital menjadi *hash* yang semula dengan melakukan dekripsi menggunakan kunci publik. Lalu, hasil dekripsi dibandingkan dengan *massage digest* dokumen. Jika sama maka dokumen tersebut sah dan jika tidak sama dokumen tersebut tidak sah. Berikut adalah contoh proses verifikasi pada e-dokumen :

1. Ubah *chiphertext* e-dokumen menjadi pesan semula yaitu dengan persamaan
   $M = C^d \bmod n$
   C = 48 26 40 56 26 58 40 58 56 25 45 62 57 40 57 26 57 49 58 56 57 56 49 45 56 70 48 26 45 39 62 55 40 85 62 55 48 79 45 70 70 45 62 58 49 56 58 87 56 70 55 25 48 79 57 85 70 56 26 25 45 26 87 79.
   d = 29

Tabel 12 Hasil Perubahan *chiphertext*

48^{29} mod 91 = 55	26^{29} mod 91 = 52	40^{29} mod 91 = 66	56^{29} mod 91 = 49
26^{29} mod 91 = 52	58^{29} mod 91 = 67	40^{29} mod 91 = 66	58^{29} mod 91 = 67
56^{29} mod 91 = 49	25^{29} mod 91 = 51	45^{29} mod 91 = 54	62^{29} mod 91 = 69
57^{29} mod 91 = 57	40^{29} mod 91 = 66	57^{29} mod 91 = 57	26^{29} mod 91 = 52
57^{29} mod 91 = 57	49^{29} mod 91 = 56	58^{29} mod 91 = 67	56^{29} mod 91 = 49
57^{29} mod 91 = 57	56^{29} mod 91 = 49	49^{29} mod 91 = 56	45^{29} mod 91 = 54
56^{29} mod 91 = 49	70^{29} mod 91 = 70	48^{29} mod 91 = 55	26^{29} mod 91 = 52
45^{29} mod 91 = 54	39^{29} mod 91 = 65	62^{29} mod 91 = 69	55^{29} mod 91 = 48
40^{29} mod 91 = 66	85^{29} mod 91 = 50	62^{29} mod 91 = 69	55^{29} mod 91 = 48
48^{29} mod 91 = 55	79^{29} mod 91 = 53	45^{29} mod 91 = 54	70^{29} mod 91 = 70
70^{29} mod 91 = 70	45^{29} mod 91 = 54	62^{29} mod 91 = 69	58^{29} mod 91 = 67
49^{29} mod 91 = 56	56^{29} mod 91 = 49	58^{29} mod 91 = 67	87^{29} mod 91 = 68
56^{29} mod 91 = 49	70^{29} mod 91 = 70	55^{29} mod 91 = 48	25^{29} mod 91 = 51
48^{29} mod 91 = 55	79^{29} mod 91 = 53	57^{29} mod 91 = 57	85^{29} mod 91 = 50
70^{29} mod 91 = 70	56^{29} mod 91 = 49	26^{29} mod 91 = 52	25^{29} mod 91 = 51
45^{29} mod 91 = 54	26^{29} mod 91 = 52	87^{29} mod 91 = 68	79^{29} mod 91 = 53

Sehingga diperoleh nilai "55 52 66 49 52 67 66 67 49 51 54 69 57 66 57 52 57 56 67 49 57 49 56 54 49 70 55 52 54 65 69 48 66 50 69 48 55 53 54 70 70 54 69 67 56 49 67 68 49 70 48 51 55 53 57 50 70 49 52 51 54 52 68 53".
Ubah hasil yang telah diperoleh menjadi karakter. Sehingga menjadi **"74B14CBC136E9B9498C191861F746AE0B2EO756FF6EC81CD1F037592F143B4D5".**
Bandingkan hasil dekripsi dengan *hash* e-dokumen. Karena hasil dekripsi sama dengan *hash* e-dokumen maka dokumen yang diverifikasi adalah dokumen yang sah.

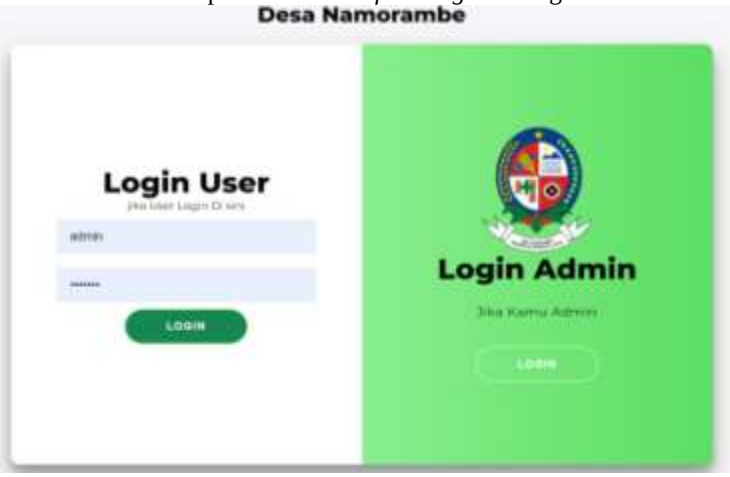
3.2 Implementasi Sistem

Hasil tampilan antarmuka merupakan gambaran hasil tampilan seluruh *form* atau menu yang terdapat pada sistem. Pada sub bab ini akan dijelaskan mengenai fungsi sistem yang telah dibangun yaitu :

1. Halaman *Login*

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Halaman *login* berfungsi sebagai proses masuk bagi setiap pengguna yaitu operator desa untuk mengakses halaman utama. Dibawah ini adalah tampilan halaman *form login* sebagai berikut :



Gambar 2 *Form Login*

2. Halaman Menu Utama
Di bawah ini adalah tampilan halaman *form* menu utama sebagai berikut :

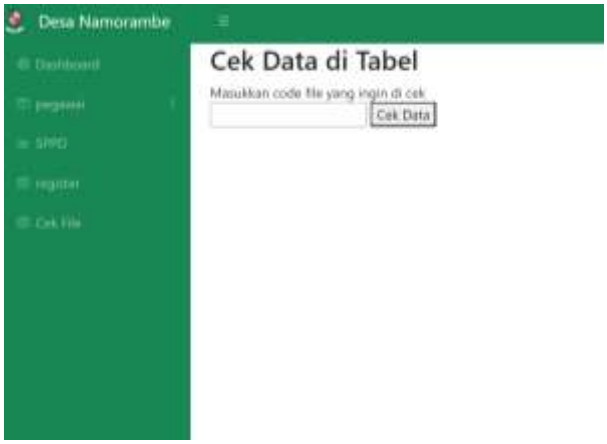
Gambar 3 *Form* Menu Utama

3. **Halaman SPPD Desa Namo Rambe**
Halaman SPPD berfungsi sebagai halaman untuk proses enkripsi serta memperoleh tanda tangan digital. Di bawah ini adalah tampilan halaman SPPD sebagai berikut :

Gambar 4 Halaman Data SPPD

4. Halaman Validasi SPPD
Halaman validasi SPPD untuk melakukan proses dekripsi guna memverifikasi tanda tangan digital serta memperoleh informasi keabsahan dari dokumen. Dibawah ini adalah tampilan validasi SPPD sebagai berikut

JURNAL SISTEM INFORMASI TGD
Volume 4, Nomor 4, Juli 2025, Hal 881-892
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi



Gambar 5 Halaman Validasi SPPD

4. KESIMPULAN

Implementasi kriptografi SHA-256 dan RSA pada tanda tangan digital untuk Surat Perintah Perjalanan Dinas (SPPD) di Kantor Desa Namo Rambe telah menunjukkan hasil yang memuaskan dalam meningkatkan keamanan dokumen. Tanda tangan digital yang diterapkan berfungsi untuk menjaga integritas serta keabsahan data, yang dilakukan dengan meringkas isi dokumen menggunakan algoritma kriptografi. Proses ini mencegah pemalsuan surat dan memastikan bahwa setiap perubahan pada dokumen dapat terdeteksi, sehingga keaslian dokumen tetap terjaga.

Dalam proses perancangan sistem tanda tangan digital, diagram Unified Modeling Language (UML) seperti use case diagram, class diagram, dan activity diagram telah digunakan untuk memodelkan sistem yang mengamankan data SPPD di Desa Namo Rambe. Penggunaan UML membantu dalam visualisasi struktur dan alur kerja sistem, memungkinkan pengembang untuk merancang sistem dengan lebih terstruktur dan terorganisir. Setiap komponen sistem dirancang untuk bekerja bersama dalam memastikan bahwa dokumen SPPD terlindungi dari manipulasi dan penyalahgunaan.

Pengujian terhadap tanda tangan digital yang diterapkan pada SPPD menunjukkan bahwa sistem tersebut memenuhi semua syarat umum tanda tangan digital, yaitu autentikasi, integritas, dan keabsahan. Implementasi algoritma SHA-256 dan RSA berhasil diterapkan dengan baik, memastikan bahwa setiap tanda tangan digital yang dihasilkan dapat diandalkan untuk mengamankan dokumen SPPD. Kesimpulannya, sistem ini efektif dalam meningkatkan keamanan dokumen di Kantor Desa Namo Rambe, memberikan perlindungan tambahan terhadap potensi ancaman digital..

UCAPAN TERIMAKASIH

Terima Kasih diucapkan kepada kedua orang tua serta keluarga yang selalu memberi motivasi, Doa dan dukungan moral maupun materi, serta pihak-pihak yang telah mendukung dalam proses pembuatan jurnal ini yang tidak dapat disebutkan satu persatu. Kiranya jurnal ini bisa memberi manfaat bagi pembaca dan dapat meningkatkan kualitas jurnal selanjutnya.

DAFTAR PUSTAKA

[1] S. Nainggolan, "Implementasi Algoritma SHA-256 Pada Aplikasi Duplicate Document Scanner," *RESOLUSI : Rekayasa Teknik Informatika dan Informasi*, vol. II, no. 5, pp. 201-213, 2022.

[2] N. P. Utomo, N. Fahriani and M. A. , "Implementasi Kriptografi Dengan Metode RSA Untuk Keamanan Data Pada Email Berbasis PHP," *Prosiding-Seminar Nasional Teknologi Informasi & Ilmu Komputer (SEMASTER)*, vol. II, no. 1, pp. 97-105, 2023.

[3] Y. Suharya and H. Widia, "Implementasi Digital Signature Menggunakan Algoritma Kriftografi Rsa Untuk Pengamanan Data Di SMK Wirakarya 1 Ciparay," *Jurnal Informatika – COMPUTING* , vol. VII, no. 1, pp. 20-29, 2020.

[4] E. C. Prabowo and I. Afrianto, "Penerapan Digital Signature Dan Kriptografi Pada Otentikasi Sertifikat Tanah Digital," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. VI, no. 2, pp. 83-90, 2017.

[5] S. T. C. Kurniawan, D. and S. , "Implementasi Kriptografi Algoritma Rivest Shamir Adleman dengan Playfair Cipher pada Pesan Teks Berbasis Android," *JOIN (Jurnal Online Informatika)*, vol. II, no. 2, pp. 102-109, 2017.

[6] N. Amalya, . S. M. S. Silalahi, D. F. Nasution, M. Sari and I. Gunawaan, "Kriptografi dan Penerapannya Dalam Sistem Keamanan Data," *JURNAL MEDIA INFORMATIKA [JUMIN]*, vol. IV, no. 2, pp. 90-93, 2023.

[7] Z. Arif and A. Nurokhman, "Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi," *JTSI*, vol. IV, no. 2, pp. 394-405, 2023.

[8] U. Wahyuningsih, M. Tahir, A. D. P. Andreani and A. Firdausi, "Analisis Proses Enkripsi Algoritma Kriptografi Modern Advanced Encryption Standard (AES)," *Jurnal Multidisplin*, vol. I, no. 2, p. 380–387, 2023.

[9] H. R. Wijaya, K. Farandi, S. Miharja and W. , "Implementasi Algoritma Aes-128 Dan Sha-256 Dalam Perancangan Aplikasi Pengamanan File Dokumen," *Jurnal TIMES*, vol. X, no. 2, pp. 80-87, 2021.

[10] A. M. Fajrin, J. R. Benedict and H. J. Kusuma, "Analisis Performa dari Algoritma Kriptografi RSA dan ElGamal dalam Enkripsi dan Dekripsi Pesan," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. VIII, no. 1, pp. 91-98, 2023.

[11] M. Z. Solihin and K. A. M., "Implementasi Kriptografi Menggunakan Metode Algoritma Rsa Pada Aplikasi Pengamanan Data Berbasis Java Desktop Untuk Ud Tirta Soeper Teloer," *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, vol. I, no. 1, pp. 351-359, 2022.

[12] S. J. Siregar, N. B. Nugroho and H. Sigalingging, "Implementasi Algoritma Kriptografi RSA (Rivest Shamir Adleman) Dalam Pengamanan Data Gaji Karyawan Di Kantor BSPJI," *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. XXII, no. 2, pp. 528-538, 2023.