semanTIK, Vol.8, No.1, Jan-Jun 2022, pp. 45-52
ISSN: 2502-8928 (Online)
DOI: 10.55679/semantik.v8i1.21608
Terakreditasi "Peringkat 4 (Sinta 4)" oleh Kemenristekdikti
Website: http://ojs.uho.ac.id/index.php/semantik

PENERAPAN KONSEP *DIGITAL SIGNATURE* TERHADAP VERIFIKASI KEASLIAN DOKUMEN TRANSKRIP NILAI MAHASISWA MENGGUNAKAN ENKRIPSI *RIVEST SHAMIR ADLEMAN*

Muh Fachrul[1], Sutardi [2], L.M Tajidun [*3], L.M Bahtiar Aksara [4]
[1,2,3,4] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail : [1]muhfachrul720@gmail.com, [2]sutardi_hapal@yahoo.com, [*3]**lmtajidun@uho.ac.id**, [4]bahtiar.aksara@uho.ac.id

Abstrak

Proses validasi dan pemberian tanda tangan dokumen transkrip nilai pada jurusan Teknik Informatika Universitas Halu Oleo mempunyai sebuah celah, celah disini yakni dapat dilakukannya sebuah pemalsuan tanda tangan dan modifikasi nilai nilai pada dokumen. Untuk mencegah hal tersebut dapat terjadi alangkah lebih baik apabila digunakannya konsep *Digital Signature* untuk proses tanda tangan dan verifikasi dokumen transkrip nilai mahasiswa. Maka dari itu dibuatlah suatu sistem berbasis web yang dapat memberikan sebuah tanda tangan digital serta melakukan verifikasi keaslian dokumen menggunakan *hashing* SHA-256 dan algoritma *Rivest Shamir Adleman*. SHA-256 merupakan algoritma *hashing* yang digunakan untuk membentuk *message digest* sebagai dasar pembentuk *digital signature* dokumen. Algoritma *Rivest Shamir Adleman* merupakan algoritma kriptografi asimetris yang menggunakan dua kunci yakni kunci publik dan kunci privat untuk proses enkripsi dan dekripsi, dalam *digital signature* sendiri kunci privat digunakan untuk membentuk tanda tangan dan kunci publik digunakan untuk proses verifikasi.

Kata kunci; *Digital Signature*, Transkrip, SHA-256, Nilai, *Rivest Shamir Adleman* (RSA)

Abstract

The process of validating and signing the student transcript at the Department of Informatics, Halu Oleo University has a flaw, the flaw here is that a forgery of signatures can be carried out as well as the modification of the values in the document. To prevent this from happening, it would be better if the Digital Signature concept was used for the signature process and verification of student transcripts. Therefore, a web-based system was created that can provide a digital signature and verify the authenticity of documents using SHA-256 hashing and the Rivest Shamir Adleman algorithm. SHA-256 is a hashing algorithm that is used to form a message digest as the basis for forming a digital signature document. The Rivest Shamir Adleman algorithm is an asymmetric cryptographic algorithm that uses two keys namely the public key and the private key for the encryption and decryption process, in the digital signature itself the private key is used to form a signature and the public key is used for the verification process.

Keywords; *Digital Signature, Transcript, SHA-256, Rivest Shamir Adleman (RSA)*

1. PENDAHULUAN

Pencetakan transkrip nilai pada Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo masih menggunakan cara manual, dokumen transkrip nilai tersebut dicetak dalam bentuk kertas dan diberikan kepada mahasiswa yang bersangkutan, berikut

mahasiswa akan meminta tanda tangan dan stempel kepada kepala jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo sebagai legalitas dan keaslian dokumen tersebut. Pada proses ini dokumen dikatakan asli apabila dokumen mempunyai stempel dan tanda tangan dari kepala jurusan, hal ini memungkinkan dokumen yang diterima mahasiswa dibuat ulang dan dimodifikasi terlebih dahulu sebelum meminta tanda tangan dan stempel jurusan.

Melihat masalah tersebut perlu dilakukan suatu tindakan pencegahan, salah satunya dengan menggunakan konsep *digital signature*, *digital signature* memiliki fungsi sebagai *marking* pada data memastikan bahwa data tersebut adalah data yang sebenarnya (tidak ada yang berubah) alhasil keaslian dari dokumen dapat dipastikan

Digital Signature bekerja dengan cara mengubah dokumen digital tersebut menjadi *message digest*, sebuah keluaran fungsi dari *hash* biasanya berbentuk simbol dan angka matematis yang kemudian dienkripsi menggunakan algoritma kriptografi *asimetris*, hasil dari enkripsi tersebutlah yang menjadi tanda tangan dari dokumen tersebut [1]. Setelah itu pada proses verifikasi keaslian dokumen dibutuhkan kunci, *file* tanda tangan dan dokumen yang bersangkutan. *File* tanda tangan dideskripsi menjadi *message digest* yang nantinya digunakan sebagai pembanding dari *message digest* keluaran fungsi *hash* dokumen. Apabila terdapat kesamaan maka *file* terbukti keasliannya tanpa dimodifikasi sama sekali.

Terdapat beberapa algoritma kriptografi *asimetris* yang dapat digunakan pada konsep *digital signature* salah satunya yaitu *Rivest Shamir Adleman* (RSA), RSA adalah salah satu teknik kriptografi dimana kunci untuk melakukan enkripsi berbeda dengan kunci untuk melakukan dekripsi. Kunci untuk melakukan enkripsi disebut sebagai kunci publik, sedangkan kunci untuk melakukan dekripsi disebut sebagai kunci privat. Orang yang mempunyai kunci publik dapat melakukan enkripsi tetapi yang dapat melakukan dekripsi hanyalah orang yang memiliki kunci privat. Kunci publik dapat dimiliki oleh sembarang orang, tetapi kunci privat hanya dimiliki oleh orang tertentu saja[1].

2. METODE PENELITIAN

2.1 Kriptografi

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci deskripsi. Deskripsi menggunakan kunci deskripsi untuk mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter [2].

2.2 *Hashing* SHA-256

Hash adalah fungsi yang secara efisien mengubah *string input* dengan panjang berhingga menjadi *string output* dengan panjang tetap yang disebut nilai *hash*, sedangkan fungsi *hash* kriptografis adalah fungsi *hash* yang memiliki beberapa sifat keamanan tambahan sehingga dapat dipakai untuk tujuan keamanan data. Umumnya digunakan pada keperluan autentikasi dan integritas data [3].

SHA-256 adalah fungsi *hash* kriptografi dengan panjang *message digest* keluaran 256 bit, merupakan fungsi *hash* yang tak membutuhkan kunci. Pesan diproses oleh blok $512 = 16 \times 32\ bit$,tiap blok membutuhkan 64 putaran.

Dalam proses komputasinya, SHA– 256 menggunakan enam fungsi logik, dimana setiap fungsi beroperasi menggunakan tiga buah *word* 32 bit ($x, y\ dan\ z$) dan keluarannya berupa sebuah word 32 bit. Berikut ini adalah fungsi-fungsi dalam SHA-256 [3]. Adapun nilai *hash* awal dari pada fungsi SHA-256 adalah sebagai berikut: [3]

$$H_1^{(0)} = 6a09e667$$
$$H_2^{(0)} = bb67ae85$$
$$H_3^{(0)} = 3c6ef372$$
$$H_4^{(0)} = a54ff53a$$
$$H_5^{(0)} = 510e527f$$
$$H_6^{(0)} = 9b05688c$$
$$H_7^{(0)} = 1f83d9ab$$
$$H_8^{(0)} = 5be0cd19$$

Nilai konstanta dalam fungsi *hashing* SHA-256 adalah sebagai berikut: [3]

```
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
```

Gambar 1 Nilai Konstanta SHA-256

Fungsi SHA-256 memiliki 8 langkah kerja adalah sebagai berikut:

1. Tambahkan *bit padding*

Pesan diisi sehingga panjangnya kongruen dengan 448, modulus 512. *Padding* 1 bit ditambahkan di akhir pesan, diikuti oleh banyak nol yang diperlukan sehingga panjang bit sama dengan 448 modulus 512.

2. Panjang *append*

Representasi panjang pesan 64 bit ditambahkan pada hasil akhirnya, langkah untuk membuat panjang pesan kelipatan 512 bit.

3. *Parsing* pesan

Pesan *padding* diuraikan menjadi *N* blok pesan 512 bit, $M(1), M(2), \dots, M(N)$, dengan menambahkan blok 64 bit.

4. Inisialisasi nilai *hash*

Nilai *hash* awal, $H(0)$ diatur, terdiri dari delapan kata 32 bit, dalam bentuk heksadesimal.

5. Mempersiapkan jadwal pesan

SHA-256 menggunakan jadwal pesan enam puluh empat kata 32 bit, kata-kata dari jadwal pesan diberi label $W0, W1, .., W63, wd$.

$$Wt = \begin{cases} M_t^{(t)}, & 0 \le t \le 15 \\ \sigma_1^{256}(W_{i-2}) + (W_{i-7}) + \sigma_0^{256}(W_{i-15}) + (W_{i-16}), & 16 \le t \le 63 \end{cases} \quad (1)$$

Dimana :

$$\sigma_1^{256}(W_{i-2}) = ((W_{i-2})ROTR\ 17) \oplus ((W_{i-2})ROTR\ 19) \oplus ((W_{i-2})SHR10) \quad (2)$$

$$\sigma_0^{256}(W_{i-15}) = ((W_{i-15})ROTR\ 17) \oplus ((W_{i-15})ROTR\ 18) \oplus ((W_{i-15})SHR3) \quad (3)$$

6. Inisialisasi

Inisialisasi delapan variabel kerja a, b, c, d, e, f, g dan h dengan nilai *hash*.

$$\begin{aligned} &(i-1)\ for\ t = 0\ to\ 63 \\ &\{ T_1 = h + \textstyle\sum_1^{(256)}(e) + Ch(e,f,g) + \\ &K_1^{(256)} + W_t \end{aligned} \quad (4)$$

$$\begin{aligned} T_2 &= \textstyle\sum_0^{256}(a) + M_{aj}(a,b,c) \\ h &= g \\ f &= e \\ e &= d + T_1 \\ d &= c \\ c &= b \\ b &= a \\ a &= T_1 + T_2 \} \end{aligned}$$

Dimana :

$$\sum_1^{(256)}(e) = (eROTR6) \oplus (eROTR11) \oplus (eROTR25) \quad (5)$$

$$\sum_0^{(256)}(a) = (aROTR2) \oplus (aROTR13) \oplus (aROTR22) \quad (6)$$

$$Ch(e,f,g) = (e \wedge f) \oplus (e \wedge g) \quad (7)$$

$$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c) \quad (8)$$

7. Menjumlahkan hasil akhir a, b, c, d, e, f, g dan h dengan inisial *hash value* $H^{(i)}$

Tabel 1 Penjumlahan inisial *hash*

$H_0^{(i)}$	$= a + H_0^{(i)}$
$H_1^{(i)}$	$= a + H_1^{(i)}$
$H_2^{(i)}$	$= a + H_2^{(i)}$
$H_3^{(i)}$	$= a + H_3^{(i)}$
$H_4^{(i)}$	$= a + H_4^{(i)}$
$H_5^{(i)}$	$= a + H_5^{(i)}$
$H_6^{(i)}$	$= a + H_6^{(i)}$
$H_7^{(i)}$	$= a + H_7^{(i)}$

8. *Output*

Setelah mengulangi langkah 1 hingga 4 sebanyak *N* kali, fungsi *hash* yang dihasilkan adalah sebagai berikut :

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} \|$$

2.3 *Rivest Shamir Adleman* (RSA)

Rivest Shamir Adleman atau RSA adalah algoritma *public key encryption* yang pertama kali dipublikasikan 1977 oleh Ron Rivest, Adi Shamir dan Leonard Adleman di MIT (*Massachusets Institute Of Technology*). Algoritma RSA melibatkan dua buah kunci dalam melakukan enkripsi yaitu *public key* dan *private key*. *Public key* dapat disebarluaskan ke berbagai pihak untuk melakukan enkripsi ataupun deskripsi. Pesan yang sudah terenkripsi dengan *public key* hanya dapat dienkripsi dengan menggunakan *private key* [4].

1. Pembangkitan Kunci

Meninjau dari pengertian algoritma *Rivest Shamir Adleman* (RSA) membutuhkan dua kunci yaitu kunci publik dan kunci privat,

berikut langkah langkah untuk membangkitkan kedua kunci tersebut:

a. Pilih dua buah bilangan prima sembarang, p dan q . Untuk memperoleh tingkat keamanan yang tinggi pilih p dan q yang berukuran besar, misalnya 1024 bit.
b. Hitung:
$$n = p * q \quad (9)$$
c. (sebaiknya $p \neq q$, sebab jika $p = q$ maka $n = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n) dimana n akan digunakan sebagai nilai untuk melakukan *modulus* pada *public* dan *private key*.
d. Hitung:
$$\varphi(n) = (p-1)(q-1) \quad (10)$$
e. Pilih bilangan *integer* e sehingga
$$1 < e < \varphi(n) \quad (11)$$
dan e adalah bilangan prima, dimana e akan digunakan sebagai *private key exponent*.
f. Cari nilai d sehingga memenuhi :
$$d = e - 1 \; mod \; \varphi(n), \text{ atau}$$
$$ed = 1 \; mod \; \varphi(n), \text{ atau} \quad (12)$$
$$ed \; mod \; \varphi(n) = 1$$

Private key terdiri dari n sebagai modulus dan e sebagai eksponen sedangkan *public key* terdiri dari n sebagai modulus dan d sebagai eksponen yang harus dirahasiakan. Nilai eksponen kunci *public* untuk RSA 1024 minimal 65537 untuk menjaga keamanannya. Hubungan antara pesan dapat dituliskan:

$$Med = M \; mod \; n \quad (13)$$

Jadi kebutuhan dari algoritma RSA sebelum proses adalah:
- p, q, dua bilangan prima yang berbeda
- $n = pq$
- e, dimana $FPB\ (\varphi(n), e) = 1; 1 < e < \varphi(n)$
- $d = e - 1 \; mod \; \varphi(n)$

2. Proses Enkripsi

Apabila kedua kunci telah didapatkan berikutnya adalah melakukan proses enkripsi, proses enkripsi menggunakan ekponansi dan modulus sehingga rumus dari proses enkripsi adalah sebagai berikut :

$$C = M^e \; (mod \; n) \quad (14)$$

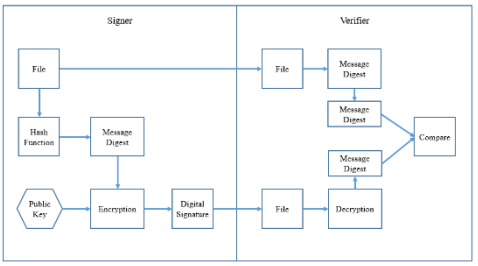
3. Proses Dekripsi

Untuk proses dekripsi, sama halnya dengan proses enkripsi yang membedakan adalah ekponen yang digunakan adalah nilai d

$$M = C^d \; (mod \; n) \quad (15)$$

2.4 *Digital Signature*

Tanda tangan merupakan sebuah tanda yang befungsi untuk memberikan kepastian keaslian dan persetujuan suatu dokumen oleh pihak pemberi tanda tangan, sama halnya dengan teknologi *digital signature* atau tanda tangan digital, bedanya *digital signature* akan memberikan legalitas terhadap dokumen elektronik terlewat dari dokumen fisik yang masih menggunakan kertas.



Gambar 2 Prinsip Kerja *Digital Signature*

Fungsi utama dari teknologi ini pada aspek keamanan kriptografi adalah *non-repudiation* atau anti penyangkalan dimana apabila dokumen *valid* maka pengirim tidak bisa menyangkal bahwa keberadaan dokumen benar dikirim oleh pengirim yang bersangkutan. [5]

2.5 *Website*

Website adalah keseluruhan halaman halaman *web* yang terdapat dari sebuah *domain* yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan [6]

2.6 PHP

PHP (akronim dari PHP: *Hypertext Preprocessor*) adalah bahasa pemograman yang berfungsi untuk membuat *website* dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file*, dan *folder*, sehingga membuat PHP bisa menampilkan konten dinamis dari sebuah *website* [7].

2.6 *Codeigniter*

Codeigniter adalah sebuah *framework* yang bersifat *open source* dan menggunakan konsep MVC (*Model*, *View*, *Controller*) untuk memudahkan *developer* atau *programmer*

dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal [8].

2.7 *Unified Modeling Language* (UML)

UML (*Unified Modelling Languange*) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang, mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah *system* [9]

Unified Modelling Languange menyediakan 8 jenis diagram yang dapat dikelompokkan berdasarkan sifat – sifatnya (statis dan dinamis), diantaranya *use case diagram*, *class diagram*, *sequence diagram*, *activity diagram* dan lainnya [10]

3. HASIL DAN PEMBAHASAN

3.1 Data

Data yang digunakan dalam penelitian ini adalah kumpulan data transkrip nilai milik mahasiswa Universitas Halu Oleo, Fakultas Teknik, jurusan Teknik informatika, data ini didapatkan dengan meminta izin kepada yang terhormat ketua Jurusan Teknik Informatika dan meminta beberapa transkrip nilai kepada pihak mahasiswa.

3.2 Penerapan Algoritma *Rivest Shamir Adleman* pada sistem

Algoritma *Rivest Shamir Adleman* memiliki tugas dalam membentuk *digital signature* dokumen dengan cara melakukan enkripsi terhadap *message digest* atau hasil *hashing* dokumen, *message digest* ini memiliki panjang yang tetap tergantung dari metode *hashing* yang digunakan pada SHA-256 sendiri panjang *message digest* adalah 64 bit. Untuk membentuk *digital signature* pertama-tama algoritma *Rivest Shamir Adleman* membentuk dua kunci yakni kunci publik dan kunci privat. Contoh proses pembentukan kunci adalah sebagai berikut:

Diketahui *message digest* dari sebuah dokumen adalah :

4dbd8328eaaa6bb0dd866ccbe82f6aae388b4fbf9 e51dac02c37dfc12eb1df4

Berikutnya penginputan dua bilangan secara acak, kedua bilangan acak ini akan menjadi nilai dari p dan q,

$p = 23$ dan $q = 37$

Kemudian dilanjutkan dengan menghitung nilai n menggunakan persamaan berikut:

$$n = p \times q$$
$$n = 23 \times 37$$
$$n = 851$$

Berikutnya mencari nilai dari m menggunakan persamaan dibawah

$$m = (p - 1)(q - 1)$$
$$m = (23 - 1)(37 - 1)$$
$$m = 792$$

Setelah mendapatkan nilai dari n dan m, berikutnya dilakukan pencari dari nilai e dan d. nilai e dan d merupakan nilai yang menjadi eksponen dalam melakukan proses enkripsi dan dekripsi dan juga merupakan pasangan dari nilai n sebagai kunci, Untuk mencari nilai e, dilakukan pemilihan bilangan yang relatif prima dari nilai m atau

$$gcd(e, 851) = 1$$

Nilai yang telah dipilih kemudian diuji kembali untuk mengetahui apakah nilai tersebut cocok dengan nilai m. Pengujian dilakukan dengan menggunakan algoritma *Euclidean*, algoritma ini merupakan algoritma yang digunakan untuk mencari FPB dari suatu nilai, Percobaan dilakukan dengan memilih nilai e adalah 5 atau $e = 5$. Untuk pengujiannya dapat dilihat sebagai berikut:

$r_0 = e$ menjadikan $r_0 = 5$,
$r_1 = m$ menjadikan $r_1 = 792$,
$q_1 = 0$,
Iterasi pertama,
$r_2 = r_0 - (q_1 \times r_1)$
$r_2 = 5 - (0 \times 792)$
$r_2 = 5$
……..
Iterasi keempat,
$r_5 = r_5 - (q_4 \times r_4)$
$r_5 = 2 - (2 \times 1)$
$r_5 = 0$

Iterasi dilakukan secara berulang hingga mendapatkan nilai akhir $r = 0$, pada kasus ini, iterasi keempat menghasilkan nilai $r = 0$. Menjadikan nilai 5 cocok dengan nilai m yakni 851.

Berikutnya mencari nilai d untuk pasangan kunci privat. Nilai d dapat ditentukan dengan menggunakan Persamaan (16), dimana nilai k akan terus bertambah hingga mendapatkan nilai bulat.

$$\frac{(1 + k \times m)}{e} \quad (16)$$

Namun perlu diperhatikan hasil dari nilai d perlu memenuhi salah satu dari persamaan:

$$d = e - 1 \; mod \; \varphi \; (n), \text{ atau}$$
$$ed = 1 \; mod \; \varphi \; (n), \text{ atau} \quad (17)$$
$$ed \; mod \; \varphi \; (n) = 1$$

Proses pencari nilai d adalah sebagai berikut, diketahui k dimulai dari 1

$k = 1$

Iterasi pertama,

$d = \frac{(1 + k \times m)}{e}$

$d = \frac{(1 + 1 \times 792)}{5}$

$d = \frac{(1 + 792)}{5}$

$d = \frac{(793)}{5}$

$d = 158{,}6$

Dikarenakan nilai dari d yang ditemukan pada iterasi pertama bukanlah bilangan bulat maka iterasi dilanjutkan dengan k bernilai 2.

Iterasi kedua,

$d = \frac{(1 + k \times m)}{e}$

$d = \frac{(1 + 2 \times 792)}{5}$

$d = \frac{(1 + 1584)}{5}$

$d = \frac{(1585)}{5}$

$d = 317$

Didapatkan hasil dari iterasi kedua adalah bilangan bulat yakni 317. Selanjutnya melakukan pengujian untuk menguji apakah nilai 317 sesuai dengan syarat persamaan.

$e \times d \; mod \; m = 1$

$5 \times 317 \; mod \; 792 = 1$

$1585 \; mod \; m = 1$

$1 = 1$

Dari pengujian telah dibuktikan bahwa nilai d = 317 sesuai dengan syarat dari $e \times d \; mod \; m = 1$. Nilai d = 317. Setelah mendapatkan nilai e dan d maka pasangan kunci publik dan kunci privat telah ditemukan dimana, kunci publik adalah 5,851 dan kunci privat adalah 317,851.

Setelah mendapatkan kedua kunci maka proses enkripsi dan dekripsi dapat dilakukan, untuk membentuk *digital signature*, proses yang digunakan adalah proses enkripsi. Proses enkripsi dalam RSA dimulai dengan memecah *plaintext* per karakter menjadi blok yang kemudian nantinya diekspo nenkan dengan nilai e pada kunci dan dimodulus dengan nilai m yang terdapat pada kunci juga.

$$c_n = ASCII(m)^e \; mod \; n \quad (18)$$

Untuk proses enkripsi pada *message digest* dengan panjang 64 bit maka, terdapat 64 iterasi. Hasil enkripsi dapat diliat dibawah ini :

Tabel 2 Tabel Enkripsi

No	Char	ASCII	Persamaan	Hasil
1	4	52	52^5 % 851	209
2	d	100	100^5 % 851	269
3	b	98	98^5 % 851	2
…	…	…	……	…..
62	d	102	102^5 % 851	410
63	f	52	52^5 % 851	209
64	4	56	56^5 % 851	318

Berdasarkan dari perulangan pada Tabel 2 didapatkan hasil enkripsi dari *message digest* adalah:

209.269.2.269.318.66.35.318.307.526.526.526.315.2.2.101.269.269.318.315.315.178.178.2.307.318.35.410.315.526.526.307.66.318.318.2.209.410.2.410.166.307.477.266.269.526.178.101.35.178.66.422.269.410.178.266.35.307.2.266.269.410.209.318

Sama halnya dengan enkripsi, proses dekripsi dilakukan beriterasi sebanyak jumlah blok *ciphertext* atau *digital signature* yang dibentuk sebelumnya. Proses dekripsi menggunakan persamaan:

$$CHAR(m)_n = c^d \; mod \; n \quad (19)$$

Untuk proses dekripsi pada *digital signature* dengan panjang per blok 64 maka, terdapat 64 iterasi. Hasil dekripsi dapat dilihat di bawah ini :

Tabel 3 Tabel Dekripsi

No	Persamaan	Hasil ASCII	Hasil
1	209^{317} % 851	52	4
2	269^{317} % 851	100	d
3	2^{317} % 851	98	b
…	……	… …	…
62	410^{317} % 851	102	d
63	209^{317} % 851	52	f
64	318^{317} % 851	56	4

Berdasarkan dari perulangan dekripsi pada Tabel 3 didapatkan hasil dekripsi dari *ciphertext* adalah sebagai berikut:

4dbd8328eaaa6bb0dd866ccbe82f6aae388b4fbf9e51dac02c37dfc12eb1df4

Berdasarkan dari hasil pengujian secara manual tersebut dapat disimpulkan bahwa proses enkripsi dan dekripsi menggunakan kunci publik 5, 851 dan kunci privat 317,851 adalah berhasil. Hasil enkripsi dapat didekripsi balik menggunakan kunci yang berbeda dengan

kunci yang digunakan untuk melakukan pengacakan.

3.3 Pengujian Kecepatan Enkripsi dan dekripsi

Pengujian ini dilakukan dengan melakukan enkripsi terhadap *message digest* menggunakan berbagai macam kunci dengan panjang dan besar nilai yang berbeda-beda. Pengujian dapat dilihat pada Tabel 4 berikut.

Tabel 4 Pengujian Kecepatan Enkripsi

Plainteks	Kunci	Waktu (Sec)
4dbd8328eaaa6 bb0dd…..	851, 5	0.0001
4dbd8328eaaa6 bb0dd.....	12319, 5	0.0001
4dbd8328eaaa6 bb0dd…..	10309291, 5	0.0002
4dbd8328eaaa6 bb0dd…..	87633047, 7	0.0003

Dari pengujian Tabel 4 dapat disimpulkan bahwa proses enkripsi tidak memakan waktu yang banyak, walaupun kunci untuk enkripsi memilki nilai yang besar, ini terjadi dikarenakan eksponen dari kunci memiliki nilai yang kecil.

Tabel 5 Pengujian Kecepatan Enkripsi

Plainteks	Kunci	Waktu (Sec)
209.307.266…	851, 371	0.0003
2735.10142….	12319, 9677	0.0364
9069556.4932972 …	10309291, 2060237	388225
48428171.230881 62...	8763304, 50065303	1.704.932 5

Untuk proses dekripsinya sendiri memiliki perbedaan dengan proses enkripsi dimana, kecepatan waktu dekripsi terbilang lebih lama berdasarkan kunci yang digunakan, semakin besar nilai dari kunci maka semakin lama pula proses dekripsi dilakukan.

3.4 Pengujian Verifikasi Sistem

Proses verifikasi merupakan salah satu proses yang terdapat dalam sistem, proses ini merupakan proses untuk mengetahui apakah dokumen telah dimodifikasi atau tidak, dalam algoritma sendiri proses ini melibatkan dekripsi. Hasil dekripsi kemudian akan di bandingkan dengan *hashing* dokumen apabila *hashing* dokumen dan hasil dekripsi berbeda maka dokumen telah dimodifikasi dan begitu juga sebaliknya. Tabel 6 merupakan tabel pengujian verifikasi dokumen dengan melibatkan dokumen transkrip yang telah dimodifikasi dan tidak dimodifikasi.

Tabel 6 Verifikasi Non Modifikasi

Konverter Word	Konverter PDF	Modifikasi	Hasil
smallpdf	*Office 2016*	Tidak	Asli
lovepdf	*Office 2016*	Tidak	Asli
simplypdf	*Office 2016*	Tidak	Asli
sodapdf	*Office 2016*	Tidak	Asli

Tabel 7 Verifikasi Modifikasi

Konverter Word	Konverter PDF	Modifikasi	Hasil
smallpdf	*Office 2016*	Iya	Palsu
lovepdf	*Office 2016*	Iya	Palsu
simplypdf	*Office 2016*	Iya	Palsu
sodapdf	*Office 2016*	Iya	Palsu

Pengujian dilakukan dengan cara mengkonversi dokumen ke dalam word terlebih dahulu kemudian dikonversi kembali ke pdf dengan diberikan perlakuan modifikasi konten dan tanpa modifikasi sama sekali.

Dari hasil pengujian telah didapatkan bahwa sistem berhasil mengenai apabila dokumen telah dimodifikasi atau tidak, walaupun dokumen telah dikonversi ke word oleh beberapa konverter yang berbeda-beda.

4. KESIMPULAN

Berdasarkan studi literatur, analisis, perancangan, implementasi dan pengujian pada sistem verifikasi keaslian dokumen transkrip nilai mahasiswa menggunakan metode *Rivest Shamir Adleman* (RSA) berbasis web ini, memiliki beberapa kesimpulan antara lain :

1. Sistem pemberian tanda tangan dan verifikasi keaslian dokumen transkrip nilai mahasiswa pada jurusan Teknik Informatika Universitas Halu Oleo berhasil dibangun dengan menggunakan metode kriptografi *Rivest Shamir Adleman* (RSA).
2. Berdasarkan hasil pengujian yang dilakukan, sistem pemberian tanda tangan dan verifikasi keaslian dokumen transkrip nilai mahasiswa pada jurusan Teknik informatika mampu membuat tanda tangan digital menggunakan proses enkripsi dan melakukan verifikasi menggunakan proses dekripsi.
3. Semakin tinggi kunci yang digunakan untuk melakukan enkripsi dan dekripsi semakin lama juga waktu yang

dibutuhkan sistem untuk melakukan proses

5. SARAN

Adapun saran yang dapat diberikan untuk pengembangan dan perbaikan sistem ini yaitu diharapkan kedepannya dapat dilakukan pengembangan dalam proses verifikasi QRCode dan pembentukan *digital signature* terhadap transkrip nilai taraf *level* fakultas, kemudian diharapkan untuk adanya integrasi data dari siakad untuk menghindari proses peng-uploadan *file* nilai mentah oleh *staff* jurusan, serta diimplementasikan nya metode yang lebih baik dari *Rivest Shamir Adleman* (RSA) untuk proses tanda tangan dan verifikasi

DAFTAR PUSTAKA

[1] T. Rahajoeningroem and M. Aria, "Studi dan Implementasi Algoritma RSA untuk pengamanan Data transkip Mahasiswa," *Maj. Ilm. Unikom*, vol. 8, no. 1, pp. 77–90, 2011, [Online]. Available: http://jurnal.unikom.ac.id/_s/data/jurnal/v08-n01/volume-81-artikel-9.pdf/pdf/volume-81-artikel-9.pdf.

[2] S. Kromodimoeljo, *Teori dan Aplikasi Kriptografi*. SPK IT Consulting, Jakarta, 2009.

[3] J. B. Awotunde *et al.*, "The cryptographic hash function SHA-256," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 64–69, 2017.

[4] I. R. Munir, "Algoritma RSA dan ElGamal," *Kriptografi*, p. 13, 2010, [Online]. Available: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Algoritma RSA.pdf.

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

[6] Yuhezifar, *Cara mudah Membangun Website interaktif Menggunakan Content Management System Joomla Edisi Revisi*. Jakarta: PT Elex Media Komputindo, 2009.

[7] T. Yuliano, "Pengenalan PHP," *Ilmu Komput.*, pp. 1–9, 2017.

[8] B. Sidik, *Framework Codeigniter*. Penerbit Informatika, Bandung, 2012.

[9] S. Dharwiyanti and R. S. Wahono, "Pengantar Unified Modeling LAnguage (UML)," *IlmuKomputer.com*, pp. 1–13, 2003, [Online]. Available: http://www.unej.ac.id/pdf/yanti-uml.pdf.

[10] I. Zufria, "Pemodelan Berbasis UML (Unified Modeling Language) dengan Strategi Teknik Orientasi Objek User Centered Design (UCD) dalam Sistem Administrasi Pendidikan Pemodelan Berbasis UML (Unified Modeling Language) dengan," *Pemodelan Berbas. UML (Unified Model. Lang. dengan Strateg. Tek. Orientasi Objek User Centered Des. dalam Sist. Adm. Pendidik.*, no. August, 2016.