Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 5 No. 4 (2021) 663 - 671 ISSN Media Elektronik: 2580-0760

Implementasi Sistem Otentikasi Dokumen Berbasis *Quick Response* (QR) *Code* dan *Digital Signature*

Antika Lorien[1], Theophilus Wellem[2]
[1,2]Teknik Informatika, Teknologi Informasi, Universitas Kristen Satya Wacana
[1]672017121@student.uksw.edu, [2]theophilus.wellem@uksw.edu*

Abstract

The authenticity and integrity of documents are essential in data exchange and communication. Digital documents must be verifiable for their authenticity and integrity by all parties that use the documents. Generally, digital documents can be authenticated by using digital signatures. This study aims to implement a document authentication system based on Quick Response (QR) code and digital signature. As the case study, the document authentication system is implemented to generate digital signatures for student's certificate documents. Furthermore, the system can also verify the authenticity of the certificate documents. Creating a digital signature requires a hash function algorithm for generating the message digest of the document. In addition, an algorithm to generate the public key and the private key used in the encryption/decryption of the message digest is also needed. The hash function utilized in this study is the Secure Hash Algorithm-256 (SHA-256), while the algorithm used for encryption/decryption is the Rivest-Shamir-Adleman (RSA) algorithm. The system is evaluated by verifying 30 student certificate documents, of which 15 of them were certificates with QR code signature generated by the system and the other 15 were certificates with QR code signature generated using a random QR code generator. The system's testing results demonstrate that the system can ensure the authenticity and integrity of the signed certificate documents to prevent document falsification. All documents that contain random QR codes were correctly identified as false documents.

Keywords: *Digital Signature*, *Quick Response* (QR) *code*, *Secure Hash Algorithm*-256 (SHA-256), Rivest-Shamir-Adleman (RSA), *Error correction.*

Abstrak

Keaslian dan integritas dari sebuah dokumen merupakan hal yang esensial dalam pertukaran data pada aplikasi teknologi informasi dan komunikasi. Oleh karena itu, suatu dokumen digital harus dapat diverifikasi keasliannya oleh pihak-pihak yang menggunakan dokumen tersebut agar dapat diketahui apakah dokumen tersebut otentik atau tidak. Secara umum, dokumen digital dapat dijaga keasliannya menggunakan metode atau teknik dalam kriptografi, yaitu tanda tangan digital (*digital signature*). Penelitian ini bertujuan untuk mengimplementasikan suatu sistem otentikasi dokumen berbasis *Quick Response* (QR) *code* dan *digital signature*. Sistem otentikasi dokumen yang diimplementasikan digunakan untuk menguji keaslian pada dokumen sertifikat mahasiswa sebagai contoh kasusnya. Dalam pembuatan *digital signature* dibutuhkan suatu fungsi *hash* untuk membuat *message digest* dari dokumen yang akan ditandatangani. Selain itu, suatu algoritma untuk melakukan enkripsi/dekripsi serta membangkitkan kunci publik/kunci privat juga dibutuhkan dalam proses pembangkitan tanda tangan digital. Penelitian ini menggunakan *Secure Hash Algorithm*-256 (SHA-256) sebagai fungsi *hash* dan algoritma Rivest-Shamir-Adleman (RSA) untuk proses enkripsi/dekripsi dan pembangkitan kunci publik/kunci privat. Pengujian sistem dilakukan terhadap 30 sertifikat, di mana 15 sertifikat mencantumkan QR *code* yang dihasilkan oleh sistem sebagai *signature*-nya dan 15 sertifikat lainnya memuat *signature* QR *code* yang dibuat menggunakan QR *code generator* lain. Dari pengujian sistem didapatkan hasil bahwa implementasi sistem otentikasi dokumen berbasis QR *code* dan *digital signature* ini dapat memastikan keaslian dan integritas dokumen sehingga mencegah pemalsuan dokumen. Semua dokumen yang mencantumkan QR *code* yang tidak dibuat atau di-*generate* oleh sistem ini berhasil diidentifikasi sebagai dokumen palsu.

Kata kunci: *Digital Signature*, *Quick Response* (QR) *code*, *Secure Hash Algorithm*-256 (SHA-256), Rivest-Shamir-Adleman (RSA), Koreksi error.

1. Pendahuluan

Dokumen adalah aset yang penting dan merupakan sumber informasi yang diperlukan oleh suatu instansi, organisasi, negara maupun individu. Jika suatu dokumen tidak dijaga dengan baik, maka kehilangan data dapat terjadi di masa yang akan datang. Seiring

Diterima Redaksi: 21-07-2021 | Selesai Revisi: 11-08-2021 | Diterbitkan Online: 20-08-2021

berkembangnya teknologi informasi, pemalsuan terhadap dokumen dapat dengan mudah dilakukan oleh pihak-pihak yang tidak berkepentingan. Bentuk pemalsuan dokumen pada umumnya dilakukan dengan memanipulasi isi dokumen dengan membuat dokumen baru menggunakan desain dan tampilan yang sama dengan aslinya. Biaya yang digunakan untuk pemalsuan dokumen juga semakin rendah, sehingga kerentanan terhadap pemalsuan meningkat. Hal ini mengakibatkan kebutuhan akan kerahasiaan informasi serta penjagaan atas keaslian dokumen semakin meningkat. Dalam perkembangan teknologi saat ini verifikasi manual menjadi kurang efektif karena membutuhkan waktu dan upaya (*effort*) yang lebih atau dengan prosedur yang cenderung rumit. Dalam penelitian ini dikembangkan suatu teknik untuk pengamanan dokumen menggunakan *digital signature* dengan mengubah signature tersebut ke dalam bentuk dua dimensi. Metode yang dikembangkan akan mengubah *digital signature* ke dalam bentuk *Quick Response* (QR) *code* [1] yang kemudian dapat dicantumkan pada dokumen fisik sebagai tanda bahwa dokumen tersebut telah ditandatangani secara digital. Proses verifikasi untuk menentukan keaslian dokumen kemudian dapat dilakukan dengan memindai QR *Code* yang terdapat pada dokumen tersebut.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan suatu sistem otentikasi dokumen yang menerapkan teknik *digital signature* dan QR *Code* untuk menjaga keaslian dokumen. Dalam implementasi sistem ini, fungsi *hash* SHA-256 dan algoritma RSA digunakan untuk membangkitkan *digital signature*. Selanjutnya, diimplementasikan juga *encoder* dan *decoder* QR *code*, dimana *encoder* berfungsi melakukan pembangkitan QR *code*, sedangkan *decoder* berfungsi untuk memindai QR *code* dan melakukan proses verifikasi terhadap dokumen sehingga dapat diketahui keaslian dari dokumen tersebut.

Penelitian ini dimulai dengan melakukan kajian terhadap beberapa penelitian terdahulu yang berhubungan. Suratma dan Azis [2] melakukan penelitian mengenai implementasi QR *code* dengan algoritma *Advanced Encryption Standard* (AES) sebagai tanda tangan digital. Hasil dari penelitian tersebut diimplementasikan pada suatu sistem otentikasi tanda tangan digital pimpinan serta verifikasi dokumen pengambilan barang yang sah.

Kurniawan, dkk. [3] melakukan perancangan sistem validasi dokumen menggunakan *digital signature* berbasis QR *code*. Pembuatan *digital signature* dalam penelitian tersebut menggunakan algoritma MD5 yang merupakan standar dalam pembuatan *digital signature*. Sistem validasi dokumen yang dirancang berfungsi untuk menerbitkan dokumen digital, melakukan validasi terhadap dokumen digital, dan meminimalisir potensi pemalsuan terhadap dokumen digital.

Prabowo dan Afrianto [4] melakukan penerapan *digital signature* dengan tujuan untuk menguji integritas dan keaslian sertifikat tanah digital. Dalam penelitian ini perubahan dokumen sertifikat serta manipulasi data oleh pihak yang tidak berkepentingan dapat dideteksi dengan *digital signature*. Algoritma *hash* untuk *digital signature* yang digunakan adalah algoritma *Secure Hash Algorithm-256* (SHA-256).

Ardhianto dan Wakhidah [5] dalam penelitiannya membahas tentang pengembangan QR *code* sebagai suatu metode otentikasi. Penelitian tersebut bertujuan untuk melakukan otentikasi kebenaran kepemilikan ijazah serta diharapkan adanya suatu metode baru atau metode tambahan yang digunakan untuk menghasilkan informasi keyakinan untuk mendeteksi keaslian ijazah.

Nuraeni, dkk. [6] dalam penelitiannya membahas tentang penggunaan QR *code* pada sertifikat elektronik untuk mempermudah pembubuhan tanda tangan digital untuk mengamankan dokumen elektronik. Pada penelitian tersebut dilakukan pengembangan tanda tangan digital untuk meningkatkan keamanan dari sistem kriptografi yaitu menggunakan algoritma RSA, algoritma AES, dan proses enkripsi 256 bit, serta algoritma Keccak.

Yuniati dan Sidiq [7] melakukan tinjauan literatur (*literature review*) terhadap beberapa jurnal penelitian terdahulu mengenai legalisasi dokumen elektronik menggunakan tanda tangan digital, kelebihan dan kekurangannya, serta diskusi tentang metode-metode yang digunakan. Hasil dari jurnal penelitian tersebut yaitu penerapan tanda tangan digital dapat dioptimalkan untuk mengatasi keterbatasan dan kendala pengesahan dokumen menggunakan cara tradisional.

Pratiwi, dkk. [8] melakukan penelitian tentang penerapan QR *code* dan algoritma AES, serta algoritma RSA untuk menjamin keamanan tanda tangan digital pada sistem informasi manajemen persuratan. Dalam penelitian tersebut proses pembentukan dan verifikasi tanda tangan digital menyediakan tingkat kepastian yang tinggi bahwa tanda tangan yang ada merupakan tanda tangan yang sah dan asli dari pemilik kunci privat.

Ihwani [9] melakukan penelitian tentang model keamanan informasi menggunakan teknik kriptografi, dan teknik kriptografi yang digunakan adalah *Digital Signature Algorithm* (DSA). Oleh karena DSA dengan fungsi *hash* tidak melakukan enkripsi terhadap *plaintext*, DSA kemudian dikombinasikan dengan RSA untuk melakukan enkripsi terhadap *plaintext*. Kajian terhadap beberapa penelitian terdahulu dirangkum pada Tabel 1.

Berbeda dengan penelitian-penelitian terdahulu, penelitian ini memanfaatkan SHA-256, RSA, dan QR *code* dalam implementasi sistem untuk otentikasi dokumen sertifikat.

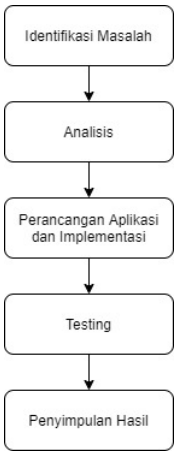
DOI: https://doi.org/10.29207/resti.v5i1.3316

Tabel 1. Penelitian Terdahulu

No	Tahun	Penulis	Algoritma yang digunakan
1.	2016	Ardhianto dan Wakhidah	Otentikasi dilakukan dengan pemrosesan citra digital pada QR *code.*
2.	2016	Ihwani	DSA yang dikombinasikan dengan RSA untuk enkripsi.
3.	2017	Suratma dan Azis	AES (untuk enkripsi/dekripsi data), QR *code* tanpa *digital signature*.
4.	2017	Prabowo dan Afrianto	*Digital signature* dengan SHA-256 dan RSA, tanpa QR *code*.
5.	2018	Pratiwi, dkk.	RSA dan AES.
6.	2020	Kurniawan, dkk.	MD5
7.	2020	Nuraeni, dkk.	RSA dan AES.
8.	2021	Yuniati dan Sidiq	Literature review mengenai legalisasi dokumen

2. Metode Penelitian

Tahapan dalam penelitian ini digambarkan pada Gambar 1.



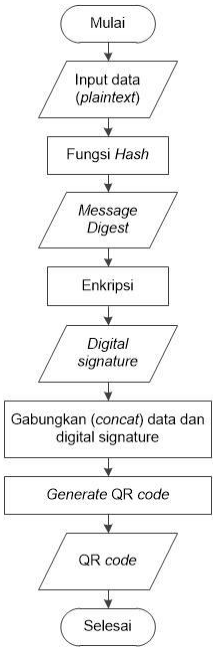
Gambar 1. Tahapan Penelitian

Berikut penjelasan Gambar 1. Proses dimulai dengan mengidentifikasi masalah, yaitu bagaimana mengimplementasikan sistem otentikasi dokumen berbasis QR *Code* dan *digital signature*. Tahap kedua yaitu menganalisis kebutuhan sistem yang akan dirancang. Ketiga yaitu perancangan aplikasi dan implementasi, dimana pada tahap ini dilakukan perancangan aplikasi berdasarkan hasil analisis kebutuhan sistem dan kemudian diimplementasikan menggunakan PHP *Programming Language*. Keempat yaitu melakukan *testing* atau pengujian sistem. Setelah merancang aplikasi maka akan dilakukan pengujian untuk memastikan aplikasi dapat bekerja dengan baik dan sesuai dengan fungsi dan rancangan awal. Tahap kelima yaitu penyimpulan hasil. Setelah proses pengujian selesai dan menghasilkan program yang sesuai dengan rancangan, dilakukan penyimpulan hasil terhadap sistem yang telah dibangun.

3. Hasil dan Pembahasan

3.1. Implementasi *Digital Signature*

Implementasi *digital signature* pada penelitian ini dimulai dengan menginputkan data yang akan ditandatangani. Selanjutnya, data tersebut digunakan sebagai input dari fungsi *hash*, yang akan menghasilkan suatu *message digest* dari data tersebut. Fungsi *hash* yang digunakan adalah SHA-256. Tahap berikutnya adalah pembangkitan kunci publik dan kunci privat, serta enkripsi terhadap *message digest* yang diperoleh sebelumnya. Enkripsi dilakukan menggunakan kunci privat dari hasil pembangkitan kunci, dan hasil enkripsi inilah yang merupakan *digital signature* dari data yang diinputkan pada sistem. Pada tahap ini digunakan algoritma Rivest-Shamir-Adleman (RSA) untuk membangkitkan kunci publik dan kunci privat serta untuk proses enkripsi terhadap *message digest*. Selanjutnya data *plaintext* yang digabungkan (*concat*) dengan *digital signature* diberikan ke QR *code generator* (*encoder*) sebagai input, sehingga akan dihasilkan sebuah *digital signature* dalam bentuk QR *code* sebagai *output*-nya. Proses pembangkitan *digital signature* dan QR *code* ini ditunjukkan pada Gambar 2. QR *code* yang dihasilkan kemudian dicantumkan pada dokumen sertifikat dan dokumen tersebut dapat dicetak.

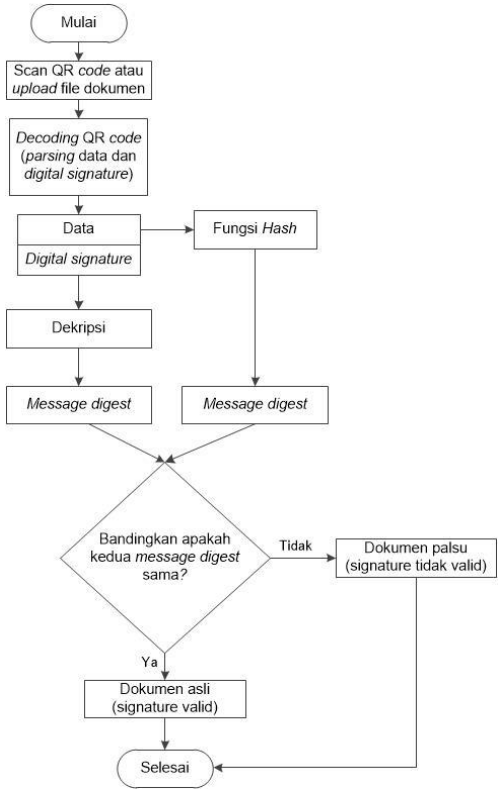


Gambar 2. Proses Pembangkitan *Digital Signature* dan *Encoding* QR *Code*

Proses verifikasi atau otentikasi dokumen pada sistem yang diimplementasikan dilakukan dengan dua cara yaitu, dengan mengunggah (*upload*) dokumen sertifikat berformat PDF pada sistem dan cara yang kedua adalah dengan melakukan pemindaian (*scan*) pada QR *code* yang tercantum pada dokumen sertifikat. Saat dokumen

DOI: https://doi.org/10.29207/resti.v5i1.3316

diunggah ke sistem atau QR *code* pada dokumen dipindai, QR *code decoder* pada sistem akan melakukan proses *decoding*. Pada proses ini, *decoder* akan memisahkan (*parsing*) bagian *digital signature* dari data yang ada pada QR *code*. Bagian data (*plaintext*) kemudian di-*hash* menggunakan fungsi *hash* SHA-256 untuk menghasilkan *message digest* dari data tersebut. Langkah berikutnya adalah melakukan dekripsi terhadap bagian *digital signature* menggunakan kunci publik untuk memperoleh kembali *message digest* yang telah dienkripsi pada saat pembuatan QR *code*. Selanjutnya, proses verifikasi dilakukan dengan cara membandingkan *message digest* dari data dengan hasil dekripsi terhadap *digital signature*. Apabila hasilnya sama, maka dapat disimpulkan bahwa dokumen tersebut asli (otentik) dan tidak terjadi perubahan atau modifikasi pada dokumen atau *digital signature*. Sebaliknya, apabila hasil perbandingan tidak sama, maka dokumen tersebut dinyatakan tidak otentik (palsu) dan telah terjadi perubahan pada dokumen atau *digital signature*. Proses *decoding* QR *code* dan verifikasi *digital signature* ditunjukkan pada Gambar 3.



Gambar 3. Proses *Decoding* QR *Code* dan Verifikasi *Digital Signature*

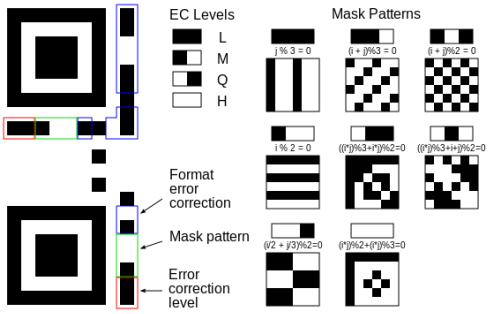
Contoh *digital signature* dan QR *code* yang dihasilkan oleh sistem otentikasi dokumen yang diimplementasikan pada penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Hasil Implementasi *Digital Signature*

Input data sertifikat pada sistem	NIM : 672017121 Nama : Antika Lorien Program Studi : Teknik Informatika Tanggal Kegiatan : 25-02-2020 Nama Kegiatan : CREATION Sie : Sekretariatan
Plaintext Signature	672017121
	s%3A267%3A%22672017121%3C%3E%11I%AF%CB%29o%1E%292C%0D%AEN%D1%DF%E7q%A3%8D8WAf%D4V%011%F2%CB%B1%F6G%A9%D0%CF%D8%AB%AE%3Ez-%DC%D8%96%C7%D7%84%E8%AA%F1%0DC%EA%97%AF%D6%03%EE%3F%FF%AF%3Dk%BBL%BC%2BR%1Fa%EB%F2%8B%96%19%F2%C8%F9wUQs%04%AC%14%3B%0B%D8%1E%A1%B2%CA%E6%0F%049%0Ar%D0%91%FDn%D7i%10%1B%BF%BA%D3%0A%C1%1B%E1%DD%C6%40%F8Cg%25%D0%CD%5Dm%D7m%3F%84%3BCa%22%29%95+%DE%DDj%7F%D5%BB%3B%16%8A%97%09%89%BE%8CZP%04k%03g%E3%CA0%5E%C0%9A%FE%17y8%CE%8F%9E%B6%F0%A5%EF%9E%2Aj%B8%A9%0F%AB%CF%5E%10p%0B%14e%BC%25%F4%21%D0%CC%D7%5C%D0e%3E%3A%C2zD%95%22%D719%F0%06%F06%8D%E6%FA%8C.Ke%00%B0r%211%CF%7E%C6-%9B%04%D1N%C4%2F%1A%E2%B4%99%1F%0D%E6n%2FY%DD%8D%0E%12o.i%28%05%23A%9A%CE%7D%22%3B
QR *Code*	

3.2. Proses *Encoding* dan *Decoding* QR *Code*

Encoding merupakan proses mengubah data ke dalam bentuk QR *code*. Proses *encoding* melalui beberapa tahap yaitu, analisis data, *encode* data, *error correction*, penambahan *reminder bits and data masking pattern*, dan pembangkitan QR *code*. Dalam proses *encoding* dibutuhkan tanda agar QR *code* yang dihasilkan dapat dibaca kembali oleh sistem dengan benar. Tanda tersebut digunakan untuk mengetahui *error correction level*, *mask pattern*, dan *format error correction* [10]. *Error correction level* merupakan tingkatan yang digunakan untuk melakukan koreksi kesalahan pada QR *code*. *Mask pattern* merupakan *grid* yang diulang untuk menutupi seluruh simbol. Tanda pada QR *code* ditunjukkan pada Gambar 4.



Gambar 4. Tanda pada QR *Code*

Proses *decoding* adalah proses menguraikan kode untuk mendapatkan informasi yang tersimpan pada QR *code* [10]. Pada sistem yang diimplementasikan, proses *decoding* membutuhkan kamera/*scanner* sebagai pemindai QR *code* atau dapat juga dilakukan dengan mengunggah *file* sertifikat pada sistem. Proses *decoding* dengan kamera pada sistem otentikasi

DOI: https://doi.org/10.29207/resti.v5i1.3316

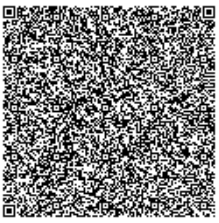
dokumen ini memanfaatkan suatu *decoder library* yaitu *webcodecam* [11]. Tahapan yang dilakukan saat proses *decoding* QR *code* yaitu identifikasi *quiet zone* dan *alignment pattern*, *decode* data, melakukan *error correction*, dan analisis data.

QR *code* dapat melakukan koreksi kesalahan dan pengembalian data pada saat pembacaan kode. Terdapat empat tingkatan koreksi kesalahan dalam QR *code* [12]. Semakin tinggi tingkat/ kemampuan koreksi kesalahan, maka semakin banyak kerusakan yang dapat diperbaiki sebelum QR *code* tersebut menjadi tidak terbaca (*decoding error*). Tingkatan koreksi kesalahan QR *code* ditunjukkan pada Tabel 3.

Tabel 3. Level Koreksi Kesalahan

Level	Kemampuan Koreksi Kesalahan
Level L (*Low*)	7%
Level M (*Medium*)	15%
Level Q (*Quartile*)	25%
Level H (*High*)	30%

Jenis QR *code* yang digunakan pada sistem otentikasi dokumen di sini adalah QR *code* dengan kemampuan koreksi kesalahan (*error correction*) level H (*high*). Level H mampu bertahan terhadap kerusakan karena pada level ini terdapat *error correction* yang mampu mengoreksi kesalahan sampai dengan 30%. *Error correction* pada QR *code* menggunakan kode Reed-Solomon yang didefinisikan pada Galois Field GF(2^8) dengan *irreducible polynomial* $x^8 + x^4 + x^3 + x^2 + 1$. Contoh QR *code* yang dihasilkan pada sistem otentikasi dokumen yang diimplementasikan ditunjukkan pada Gambar 5. Versi QR *code* yang dihasilkan adalah versi 29. QR *code* versi 29 dengan *error correction* level H mampu menampung data sebesar 5608 bits.



Gambar 5. Hasil QR *Code*

3.3. Algoritma SHA-256

Pada penelitian ini, fungsi *has*h SHA-256 digunakan untuk menghasilkan *message digest* dari sebuah pesan (*message*) atau data. Contoh *message digest* yang dihasilkan dari data pada dokumen sertifikat ditunjukkan pada Tabel 4.

Tabel 4. Hasil *Hash* SHA-256

Data	*Message Digest (Hex Format)*
672017121	a394d5ec5ded5d271dfc68ac17bb16e32f2e9988 da7db10755fa2276cdc7fc4a

3.4. Algoritma RSA

Pada penelitian ini, algoritma RSA digunakan untuk melakukan enkripsi terhadap *message digest* yang merupakan hasil dari penerapan fungsi *hash* SHA-256 pada data *plaintext*. Enkripsi terhadap *message digest* dilakukan menggunakan kunci privat sehingga menghasilkan suatu *digital signature* karena kunci privat hanya diketahui oleh pemegang kunci. Selain untuk melakukan enkripsi, algoritma ini juga digunakan untuk melakukan dekripsi *digital signature* pada proses verifikasi data. Proses dekripsi dilakukan menggunakan kunci publik.

Langkah-langkah algoritma RSA dalam pembuatan *digital signature* adalah sebagai berikut [4]. Langkah pertama yaitu pembentukan kunci publik dan kunci privat. Dalam pembentukan kunci terdapat beberapa tahap yang harus dilakukan yaitu: menentukan dua bilangan prima secara acak, bilangan prima tersebut disebut p dan q. Kemudian akan dihitung nilai n yang merupakan hasil perkalian dari kedua bilangan prima tersebut (1). Selanjutnya, dihitung nilai φ untuk menentukan banyak bilangan $1,2,3, \dots, n$ yang relatif prima dengan n (2). Berikutnya ditentukan nilai e, dimana e merupakan kunci publik (3). Setelah itu akan ditentukan nilai d, dimana nilai d merupakan kunci privat (4).

$$n = p * q \quad (1)$$

$$\varphi = (p-1)(q-1) \quad (2)$$

Bilangan bulat $1 < e < \varphi$ yang relatif prima dengan nilai φ (3)

$$d.e = 1 \bmod \varphi \quad (4)$$

Langkah kedua dalam pembuatan *digital signature* menggunakan algoritma RSA yaitu melakukan proses enkripsi berdasarkan rumus (5) dan langkah yang terakhir adalah melakukan proses dekripsi berdasarkan rumus (6).

$$C = P^d \bmod n \quad (5)$$

$$P = C^e \bmod n \quad (6)$$

dengan C adalah *ciphertext*, P adalah *plaintext*, d adalah kunci privat, dan e adalah kunci publik

Contoh penerapan *digital signature* menggunakan algoritma RSA dapat dijelaskan sebagai berikut. *Message* yang digunakan pada contoh ini diambil dari *message digest* pada Tabel 4. Langkah pertama yaitu pembuatan kunci publik dan kunci privat. Pembuatan kedua kunci tersebut dilakukan dengan memilih bilangan prima acak, yaitu 73 dan 67. Selanjutnya yaitu menentukan nilai n dengan n = 73 x 67 = 4891 (1). Setelah itu menghitung nilai φ dengan φ = (73-1) (67-1) = 4752 (2). Selanjutnya dapat ditentukan nilai e yaitu 7 (3) dan nilai d adalah 679 (4). Dari perhitungan

DOI: https://doi.org/10.29207/resti.v5i1.3316

tersebut maka dapat diketahui bahwa kunci privatnya adalah 679 dan kunci publiknya adalah 7.

Langkah kedua yaitu proses enkripsi. Proses enkripsi dengan algoritma RSA menggunakan nilai e dan n yang telah dibangkitkan sebelumnya. Langkah selanjutnya yaitu melakukan konversi *message digest* yang dihasilkan dari proses *hash* ke dalam kode ASCII, kemudian mengambil hasil konversi pada blok P_1, P_2, dan P_3. Hasil konversi *message digest* ke dalam kode ASCII dapat dilihat pada Tabel 5.

Tabel 5. Hasil Konversi ke Dalam Kode ASCII

Message Digest (Hex Format)	Dalam Kode ASCII
a394d5ec5ded5	**97 51 57** 52 100 53 101 99 53 100 101 100 53
d271dfc68ac17	100 50 55 49 100 102 99 54 56 97 99 49 55
bb16e32f2e998	98 98 49 54 101 51 50 102 50 101 57 57 56
8da7db10755fa	56 100 97 55 100 98 49 48 55 53 53 102 97
2276cdc7fc4a	50 50 55 54 99 100 99 55 102 99 52 97

Misalkan *message digest* yang akan dienkripsi dengan kunci privat adalah a39 dengan kode ASCII 97 51 57. Proses enkripsi dapat dilihat pada Tabel 6.

Tabel 6. Hasil Konversi ke Dalam Kode ASCII

Kunci Privat (d, n)	d = 679 n = 4891
Rumus Enkripsi	$C_i = P^d \bmod n$
Proses Enkripsi	$C_1 = 97^{679} \bmod 4891 = 633$ $C_2 = 51^{679} \bmod 4891 = 2326$ $C_3 = 57^{679} \bmod 4891 = 4617$
Hasil Enkripsi dengan pemisah titik (.)	633.2326.4617

Langkah ketiga yaitu proses dekripsi. Proses dekripsi menggunakan algoritma RSA membutuhkan nilai e dan n yang telah dibangkitkan sebelumnya. Proses dekripsi dilakukan dengan mengubah kembali blok *ciphertext* (C_i) menjadi blok *plaintext* (P_i). Proses dekripsi ditunjukkan pada Tabel 7.

Tabel 7. Proses Dekripsi dengan Algoritma RSA

Kunci Publik (e, n)	e = 7 n = 4891
Rumus Dekripsi	$P_i = C^e \bmod n$
Proses Dekripsi	$P_1 = 633^7 \bmod 4891 = 97$ $P_2 = 2326^7 \bmod 4891 = 51$ $P_3 = 4617^7 \bmod 4891 = 57$
Hasil Dekripsi	97 51 57
Hasil Dekripsi *(Hex Format)*	**a39**

Dari hasil dekripsi pada Tabel 7 dapat disimpulkan bahwa dokumen telah terbukti keasliannya, karena *message digest* dari proses dekripsi hasilnya sama dengan *message digest* hasil dari proses *hash*.

3.5. Implementasi Sistem

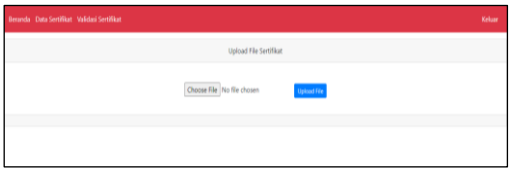
Pada sistem otentikasi dokumen berbasis QR *code* dan *digital signature* yang dirancang terdapat fitur-fitur utama yaitu, input data sertifikat, pembuatan (*generate*) QR *code*, verifikasi dokumen (sertifikat) dengan kamera/*scanner*, dan verifikasi dokumen dengan *upload* file sertifikat dalam format PDF.

Pembuatan QR *code* dilakukan dengan cara menekan tombol *Generate*. Ketika QR *code* berhasil dibuat, maka akan ditampilkan notifikasi QR *code* sukses di-*generate*, kemudian QR *code* akan langsung terinput pada halaman cetak sertifikat. Pada halaman cetak sertifikat akan ditampilkan sertifikat mahasiswa yang telah dibubuhi dengan tanda tangan digital berbentuk QR *code*, pada halaman ini *user* dapat men-*download file* sertifikat dalam format PDF. Halaman cetak sertifikat dapat dilihat pada Gambar 6.

Gambar 6. Halaman Cetak Sertifikat

Selanjutnya, halaman verifikasi sertifikat berfungsi untuk melakukan verifikasi sertifikat mahasiswa agar dapat diketahui sertifikat tersebut asli atau tidak. Pada halaman ini terdapat dua cara untuk melakukan validasi dokumen: 1) validasi dengan cara memindai QR *code* menggunakan kamera yang disediakan pada sistem, dan 2) validasi dengan cara upload file sertifikat mahasiswa dalam format PDF. Gambar 7 merupakan antarmuka halaman verifikasi dengan cara memindai QR *code* menggunakan kamera dan Gambar 8 merupakan antarmuka halaman verifikasi dengan cara mengunggah *file* sertifikat.

Gambar 7. Halaman *Scan* QR *Code*



Gambar 8. Halaman *Upload File*

Hasil verifikasi dokumen pada sistem otentikasi ini akan menunjukkan bahwa dokumen yang diverifikasi merupakan dokumen asli atau palsu. Hal tersebut diketahui melalui verifikasi terhadap *digital signature* yang ada dalam QR *code* pada dokumen sertifikat

DOI: https://doi.org/10.29207/resti.v5i1.3316

tersebut. Apabila dokumen terverifikasi asli, maka sistem menampilkan notifikasi bahwa dokumen tersebut asli dan apabila dokumen tersebut palsu, maka sistem akan menunjukkan notifikasi bahwa dokumen tersebut palsu.

3.6. Pengujian Fungsionalitas Sistem

Metode yang digunakan dalam pengujian ini adalah metode pengujian *black box* yang berfokus pada fungsionalitas semua tombol dan fitur pada setiap halaman. Pengujian *black box* dilakukan untuk menguji apakah respon pada sistem sudah sesuai dengan skenario yang seharusnya dan untuk mengetahui apakah masih terdapat kesalahan dalam sistem. Pengujian pertama dilakukan oleh peneliti sendiri dengan menguji sistem ini sesuai dengan skenario pengujian sistem. Selanjutnya lima pengguna juga melakukan pengujian dengan langkah yang sama dengan yang dilakukan oleh peneliti. Hasil pengujian sistem menunjukkan bahwa sistem telah berfungsi sesuai dengan yang diharapkan dan sesuai dengan skenario, sehingga dapat dikatakan berhasil. Hasil pengujian sistem dapat dilihat pada Tabel 8 dan Tabel 9.

3.7. Pengujian menggunakan Dokumen

Pada pengujian ini dilakukan verifikasi terhadap 30 dokumen sertifikat yang telah dicantumkan QR *code*, di mana 15 sertifikat mencantumkan QR *code* yang dihasilkan oleh sistem sebagai *signature*-nya dan 15 sertifikat lainnya memuat QR *code* yang dibuat dengan QR *code generator* lain sebagai *signature*-nya. Tabel 10 menunjukkan contoh verifikasi dokumen dengan *signature* QR *code* yang di-*generate* oleh sistem otentikasi dokumen yang telah diimplementasikan. Hasil verifikasi dokumen tersebut menunjukkan bahwa dokumen terbukti asli dan tidak terdapat manipulasi terhadap dokumen. Pengujian dilakukan terhadap 15 sertifikat dan semua dokumen tersebut terverifikasi asli atau valid

Tabel 8. Hasil Pengujian Sistem

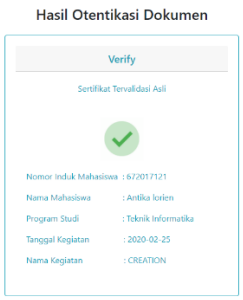
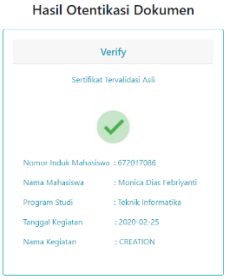
No.	Menu	Skenario	Hasil yang diharapkan	1	2	3	4	5	6
1.	Login	Masukkan *username* dan *password*	Login berhasil dan menampilkan halaman Beranda.	V	V	V	V	V	V
		• *Username* yang dimasukkan salah. • *Password* yang dimasukkan salah.	• Menampilkan pesan kesalahan *username* tidak terdaftar. • Menampilkan pesan kesalahan *password* salah.	V	V	V	V	V	V
2.	Beranda	Klik tombol Beranda.	Menampilkan halaman Beranda.	V	V	V	V	V	V
3.	Data Sertifikat	Klik tombol Data Sertifikat.	Menampilkan halaman Data Sertifikat.	V	V	V	V	V	V
4.	Cari	Tuliskan berdasarkan data yang akan dicari.	Menampilkan hasil pencarian data.	V	V	V	V	V	V
5.	Tambah data	• Klik tombol Tambah Data. • Inputkan data sertifikat pada *form* yang sudah tersedia dan klik tombol Tambah Data.	• Menampilkan *form* tambah data. • Data berhasil ditambahkan.	V	V	V	V	V	V

Tabel 9. Hasil Pengujian Sistem

No.	Menu	Skenario	Hasil yang diharapkan	1	2	3	4	5	6
1.	Delete	Klik tombol Delete.	Data berhasil dihapus.	V	V	V	V	V	V
2.	Generate QR Code	Klik tombol Generate QR Code.	QR *code* berhasil di-*generate*, setelah itu muncul tombol Cetak.	V	V	V	V	V	V
3.	Cetak Sertifikat	Klik tombol Cetak.	Menampilkan halaman sertifikat mahasiswa.	V	V	V	V	V	V
4.	Download Sertifikat	Klik tombol Download Sertifikat.	Sertifikat berhasil di-*download* dalam format PDF.	V	V	V	V	V	V
5.	Validasi Sertifikat	Klik tombol Validasi Sertifikat.	Menampilkan halaman validasi sertifikat dan terdapat dua metode yaitu validasi dengan *scan* QR *code* atau validasi dengan *upload file* sertifikat.	V	V	V	V	V	V
6.	Validasi dengan *scan* QR code	• Klik tombol Scan QR Code. • Arahkan QR *code* ke kamera.	• Menampilkan halaman Scan QR Code. • Proses validasi berhasil dan menampilkan hasil bahwa sertifikat tersebut asli atau palsu.	V	V	V	V	V	V
7.	Validasi dengan *upload file* sertifikat.	• Klik tombol Upload File. • *Upload file* sertifikat kemudian klik tombol Upload File.	• Menampilkan halaman Upload File. • Proses validasi berhasil dan menampilkan hasil bahwa sertifikat tersebut asli atau palsu.	V	V	V	V	V	V

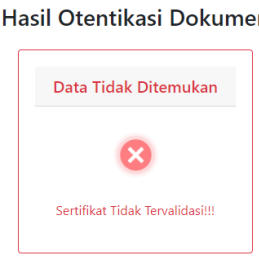
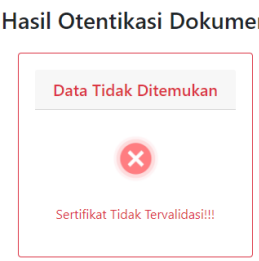
DOI: https://doi.org/10.29207/resti.v5i1.3316

Tabel 10. Contoh Verifikasi Dokumen (valid)

Verifikasi Dokumen	Hasil Verifikasi
	
	

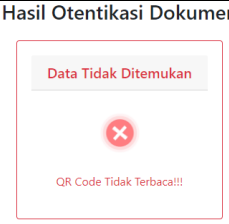
Selanjutnya dilakukan pengujian terhadap dokumen sertifikat dengan *signature* QR *code* yang bukan dihasilkan (di-*generate*) oleh sistem yang diimplementasikan. Sebanyak 15 dokumen sertifikat digunakan dalam pengujian ini dan hasilnya menunjukkan bahwa semua dokumen tersebut terdeteksi palsu atau tidak valid. Tabel 11 menunjukkan contoh verifikasi dokumen sertifikat yang mencantumkan *signature* QR *code* dari luar sistem.

Tabel 11. Verifikasi Dokumen (tidak valid)

Verifikasi Dokumen	Hasil Verifikasi
	
	

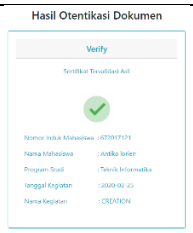
Pengujian juga dilakukan menggunakan QR *code* yang rusak. Tabel 12 menunjukkan hasil verifikasi dokumen dengan QR *code* yang rusak (dengan perkiraan kerusakan lebih dari 30%). Hasil dari pengujian tersebut menunjukkan bahwa dokumen akan ditolak.

Tabel 12. Verifikasi Dokumen dengan QR *Code* yang rusak

Verifikasi Dokumen	Hasil Verifikasi
	

Seperti yang telah disebutkan sebelumnya, sistem otentikasi dokumen yang diimplementasikan menghasilkan QR *code* dengan level koreksi kesalahan *High*. Pengujian level koreksi kesalahan dilakukan dengan membuat QR *code* untuk data yang sama, tetapi level koreksi kesalahannya berbeda (*High* dan *Low*). Kemudian kedua QR *code* tersebut dirusak (dalam hal ini dikotori dengan coretan) dan dilakukan pengujian apakah QR *code* tersebut masih bisa terbaca atau tidak. Hasil pengujian menunjukkan bahwa QR *code* yang kotor dengan level koreksi kesalahan *High* masih dapat terverifikasi oleh sistem, sedangkan QR *code* yang kotor dengan level koreksi kesalahan *Low* tidak dapat terbaca oleh sistem. Hasil pengujian tersebut ditunjukkan pada Tabel 13.

Tabel 13. Verifikasi Level Koreksi Kesalahan QR *Code*

Level Koreksi Kesalahan	Verifikasi Dokumen	Hasil Verifikasi
H (*high*)		
L (*low*)		QR *code* tidak terbaca

4. Kesimpulan

Penelitian ini membahas metode untuk menjaga keaslian (otentikasi) dokumen menggunakan *digital signature* dan *Quick Response* (QR) *code*. Metode ini

DOI: https://doi.org/10.29207/resti.v5i1.3316

kemudian diimplementasikan pada sebuah sistem otentikasi dokumen (sertifikat) yang dilengkapi dengan QR *code encoder* dan *decoder*. Sistem otentikasi dokumen ini mempunyai fitur-fitur utama yaitu: membangkitkan *digital signature* dan mengubahnya dalam bentuk QR *code*, serta mencetak dokumen dan verifikasi dokumen. Pengujian sistem menunjukkan bahwa sistem otentikasi dokumen berbasis QR *code* dan *digital signature* yang diimplementasikan pada penelitian ini dapat menjamin keaslian dokumen yang telah ditandatangani atau dibubuhi dengan QR *code*. Implementasi *digital signature* dengan QR *code encoder* dan *decoder* bermanfaat untuk menjaga keaslian dokumen, sehingga dokumen tidak dapat dipalsukan kecuali kunci privat yang digunakan untuk membuat *digital signature* diketahui oleh pihak lain.

Daftar Rujukan

[1] ISO/IEC, "ISO/IEC 18004:2015 Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification," 2015. https://www.iso.org/standard/62021.html.

[2] A. G. P. Suratma and A. Azis, "Tanda Tangan Digital Menggunakan QR Code dengan Metode Advanced Encryption Standard," *Techno*, vol. 18, no. 1, pp. 59–68, 2017, doi: 10.30595/techno.v18i1.1482.

[3] R. Kurniawan, R. F. Sari, and N. Azizah, "Sistem Validasi Keaslian Dokumen Digital Berbasis QR-Code," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 321–327, 2020, [Online]. Available: http://www.jurnal.una.ac.id/index.php/jurti/article/viewFile/1722/1468.

[4] E. C. Prabowo and I. Afrianto, "Penerapan Digital Signature dan Kriptografi pada Otentikasi Sertifikat Tanah Digital," *J. Ilm. Komput. dan Inform.*, vol. 6, no. 2, pp. 83–90, 2017, doi: 10.34010/KOMPUTA.V6I2.2481.

[5] E. Ardhianto and N. Wakhidah, "Pengembangan Metode Otentikasi Keaslian Ijasah dengan Memanfaatkan Gambar QR Code," *Transformatika*, vol. 13, no. 2, pp. 35–41, 2016, doi: 10.26623/transformatika.v13i2.325.

[6] F. Nuraeni, Y. H. Agustin, D. Kurniadi, and I. D. Ariyanti, "Implementasi Skema QR-Code dan Digital Signature menggunakan Kombinasi Algoritma RSA dan AES untuk Pengamanan Data Sertifikat Elektronik," in *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 12*, 2020, pp. 43–52.

[7] T. Yuniati and M. F. Sidiq, "Literature Review: Legalisasi Dokumen Elektronik Menggunakan Tanda Tangan Digital sebagai Alternatif Pengesahan Dokumen di Masa Pandemi," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 1058–1069, 2020, doi: 10.29207/resti.v4i6.2502.

[8] B. P. Pratiwi, M. R. Pratama, and T. A. Cahyanto, "Implementasi Algoritma AES (Advanced Encryption Standard) dan RSA (Rivest Shamir Adleman) untuk Pengamanan Surat di Humanika," Universitas Muhammadiyah Jember, 2019.

[9] M. Ihwani, "Model Keamanan Informasi Berbasis Digital Signature dengan Algoritma RSA," *CESSJ (Journal Comput. Eng. Syst. Sci.*, vol. 1, no. 1, pp. 15–20, 2016, doi: 10.24114/cess.v1i1.4037.

[10] E. F. Nurdiansyah and I. Afrianto, "Implementasi QR Code sebagai Tiket Masuk Event dengan Memperhitungkan Tingkat Koreksi Kesalahan," *J. Teknol. dan Inf.*, vol. 7, no. 2, pp. 25–44, 2018, doi: 10.34010/jati.v7i2.491.

[11] T. András, "WebCodeCamJS," 2018. https://github.com/andrastoth/webcodecamjs.

[12] M. L. Sholeh and L. A. Muharom, "Smart Presensi Menggunakan QR-Code dengan Enkripsi Vigenere Cipher," *Limits J. Math. Its Appl.*, vol. 13, no. 2, pp. 31–44, 2016, doi: 10.12962/j1829605X.v13i2.1933.

DOI: https://doi.org/10.29207/resti.v5i1.3316