Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 3, June 2024, pp. 675-684

DOI: https://doi.org/10.52436/1.jutif.2024.5.3.1426
p-ISSN: 2723-3863
e-ISSN: 2723-3871

IMPLEMENTATION OF RSA AND AES-128 SUPER ENCRYPTION ON QR-CODE BASED DIGITAL SIGNATURE SCHEMES FOR DOCUMENT LEGALIZATION

Fitri Nuraeni[*1], Dede Kurniadi[2], Diva Nuratnika Rahayu[3]

[1,2,3]Jurusan Ilmu Komputer, Institut Teknologi Garut, Indonesia
Email: [1]fitri.nuraeni@itg.ac.id, [2]dede.kurniadi@itg.ac.id, [3]1906050@itg.ac.id

(Article received: September 26, 2023; Revision: October 04, 2023; published: May 18, 2024)

Abstract

Maintaining the confidentiality and integrity of electronic documents is essential in the modern digital age. In the contemporary digital world, digital signatures are essential for safeguarding and legalizing electronic documents. The current issue, however, goes beyond digital signatures and instead centers on enhancing security and data integrity. Therefore, RSA and AES-128 super-encryption is required in QR-code-based digital signature techniques for document legalization. This research stage entails constructing a super encryption algorithm, testing it experimentally for security and performance, and designing a digital signature system using RSA and AES-128 super encryption. The results of this research show that the use of RSA and AES super encryption has been proven to have better performance in data security, where the encryption and decryption process time is relatively close to the RSA encryption time, and the comparison of entropy values is better than RSA and AES-128. So, the combination of Super RSA and AES-128 encryption can increase the security level of electronic documents and reduce the risk of hacking. Moreover, the proposed QR-code-based digital signature scheme is also very efficient regarding file size and processing time.

Keywords: *data integrity, data security, encryption performance, legalization, super encryption*.

IMPLEMENTASI ENKRIPSI SUPER RSA DAN AES-128 PADA SKEMA TANDA TANGAN DIGITAL BERBASIS QR-CODE UNTUK LEGALISASI DOKUMEN

Abstrak

Menjaga kerahasiaan dan integritas dokumen elektronik sangat penting di era digital modern. Di dunia digital kontemporer, tanda tangan digital sangat penting untuk menjaga dan melegalkan dokumen elektronik. Namun permasalahan yang ada saat ini lebih dari sekedar tanda tangan digital, melainkan berpusat pada peningkatan keamanan dan integritas data. Oleh karena itu, super-enkripsi RSA dan AES-128 diperlukan dalam teknik tanda tangan digital berbasis kode QR untuk legalisasi dokumen. Tahap penelitian ini memerlukan pembuatan algoritma enkripsi super, mengujinya secara eksperimental untuk keamanan dan kinerja, dan merancang sistem tanda tangan digital menggunakan enkripsi super RSA dan AES-128. Hasil penelitian menunjukkan bahwa penggunaan super enkripsi RSA dan AES terbukti mempunyai kinerja yang lebih baik dalam keamanan data, dimana waktu proses enkripsi dan dekripsi relatif dekat dengan waktu enkripsi RSA, serta perbandingan nilai entropi yang lebih baik daripada RSA dan AES-128. Jadi, kombinasi enkripsi Super RSA dan AES-128 mampu meningkatkan tingkat keamanan dokumen elektronik dan mengurangi risiko peretasan. Selain itu, skema tanda tangan digital berbasis kode QR yang diusulkan juga sangat efisien dalam hal ukuran file dan waktu pemrosesan.

Kata kunci: *integritas data, keamanan data, kinerja enkrips, legalisasi, super enkripsi*.

1. PENDAHULUAN

Data merupakan aset berharga dalam sebuah organisasi atau perusahaan yang dapat digunakan untuk memutuskan suatu kebijakan, melakukan aksi-aksi strategis, atau mengambil keputusan bisnis yang tepat [1]. Semakin banyak data yang disimpan dan ditransmisikan secara digital, semakin besar kemungkinan muncul ancaman yang berisiko terhadap keamanan dan integritas data[2][3][4][5].

Ancaman tersebut dapat berupa serangan siber, pencurian dan pemalsuan data, penyebaran virus, dan penggunaan sistem secara ilegal[6]. Kemudian keamanan data merujuk pada upaya untuk melindungi data dari kerusakan, kehilangan, modifikasi, akses dan penggunaan oleh pihak yang tidak berwenang[7]. Sedangkan integritas data merujuk pada keotentikan atau keaslian data dan

memastikan bahwa tidak terjadi perubahan pada data oleh pihak yang tidak berwenang[8].

Untuk meningkatkan keamanan dan integritas data dapat menggunakan kriptogtafi, yaitu ilmu dan seni menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dipahami lagi maknanya[9]. Dokumen yang diterbitkan oleh suatu instansi akan diakui dan sah untuk berbagai keperluan jika telah disahkan, seperti tanda tangan dan stempel [10]. Teknik otentikasi dokumen digital diperlukan karena tanda tangan dan stempel manusia tidak dapat digunakan pada dokumen digital[11]. Penggunaan tanda tangan digital yang tak terbantahkan menjadi salah satu teknik pengelolaan data yang digunakan untuk memastikan integritas dan kemampuan audit yang tinggi[12].

Kriptografi kunci asimetris adalah metode yang digunakan oleh tanda tangan digital untuk melindungi data yang terdapat dalam dokumen yang dilampirkan, salah satunya yaitu algoritma *Rivest Shamir Adleman* (RSA) [13]. Kemudian penggunaan *QR-Code* dapat mempermudah pembubuhan tanda tangan digital yang mempunyai kode yang cukup panjang, adapun proses verifikasi sertifikat menjadi lebih sederhana karena cukup menggunakan *QR-Code Reader* untuk mendapatkan kode tanda tangan digital dari sertifikat tersebut[14].

Penggunaan tanda tangan digital yang efisien diperlukan untuk memastikan integritas data, keaslian, dan non-penyangkalan dengan keamanan informasi-teoretis. Namun, kemunculan komputer kuantum mengharuskan pengembangan sistem keamanan yang andal untuk tanda tangan elektronik pada dokumen [15]. Super Enkripsi dapat menjadi salah satu upaya dalam meningkatkan keamanan tanda tangan digital [16], yaitu melakukan enkripsi pesan lebih dari satu kali dengan algoritma berbeda.

Kinerja super-enkripsi yang baik memiliki waktu pemrosesan yang relatif sama dengan skema enkripsi biasa, sehingga perlu membandingkan waktu proses enkripsi dan dekripsi keduanya [17]. Selain kriteria waktu proses, super-enkripsi harus dapat menghasilkan *ciphertext* yang lebih acak, sehingga tidak dapat dipahami[18].

Oleh karena itu, pada penelitian ini digunakan fungsi MD5 untuk proses *hashing* data pada dokumen, karena algoritma ini memiliki kecepatan yang lebih unggul dibanding fungsi hash lainnya[19]. Sedangkan untuk proses enkripsi pada skema tanda tangan digital ini menggunakan algoritma RSA dan AES-128 untuk mendapatkan superenkripsi yang kuat namun relatif waktu yang masih sama dengan enkripsi 1 algoritma saja.

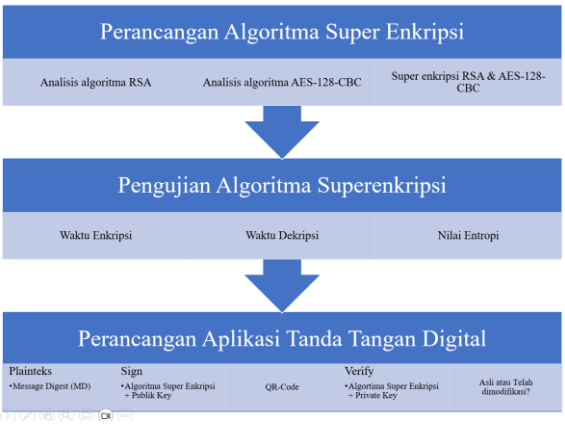
Untuk implementasi dari tanda tangan digital ini selanjutnya dibangun sebuah aplikasi tanda tangan digital berbasis QR-Code yang diharapkan dapat membantu menjaga kemanan dan integritas data pada dokumen digital serta membantu mencegah tindakan manipulasi dan perubahan data.

2. METODE PENELITIAN

Pada penelitian ini dilakukan tahapan-tahapan seperti yang disajikan pada gambar 1 dibawah ini, dimana proses utamanya dibagi dua tahapan yaitu 1) perancangan algoritma super enkripsi dan pengujiannya; dan 2) perancangan aplikasi tanda tangan digital berbasis QR-Code yang menerapkan super enkripsi RSA dan AES-128.

Pada tahapan pertama dimulai dengan melakukan studi literatur untuk mendapatkan data mengenai algoritma RSA dan AES-128, super enkripsi, pengujian kinerja algoritma kriptografi dan skema tanda tangan digital, berdasarkan hasil penelitian terdahulu.

Selanjutnya mulai merancangan algoritma super enkripsi yang menggabungkan proses enkripsi menggunakan RSA kemudian dilanjutkan enkripsi AES 128 bit menggunakan mode operasi *Cipher Block Chaining* (CBC). Kemudian algoritma super enkripsi ini diuji pada 20 plainteks dengan kriteria pengujian yaitu waktu proses enkripsi, dekripsi dan nilai entropi.



Gambar 1. Tahapan Penelitian

Tahapan kedua pada penelitian ini adalah perancangan aplikasi tanda tangan digital sesuai dengan fase-fase pada perancangan perangkat lunak *Rational Unified Process* (RUP). Fase pertama yaitu *inception*, pada fase ini aktivitas yang dilakukan adalah identifikasi proses bisnis, menentukan spesifikasi sistem dan identifikasi aktor. Fase kedua yaitu *elaboration*, pada fase ini aktivitas yang dilakukan adalah merancang use case diagram, *activity diagram*, *sequence diagram*, *class diagram*, struktur menu dan antarmuka sistem. Fase ketiga yaitu *construction*, pada fase ini aktivitas yang dilakukan adalah implementasi perancangan ke dalam bahasa pemrograman. Fase keempat yaitu *transition*, pada fase ini aktivitas yang dilakukan adalah pengujian *black box*.

2.1. Tanda Tangan Digital

Tanda tangan digital adalah suatu mekanisme otentikasi yang memungkinkan pengirim pesan untuk menambahkan sebuah kode yang berperan sebagai

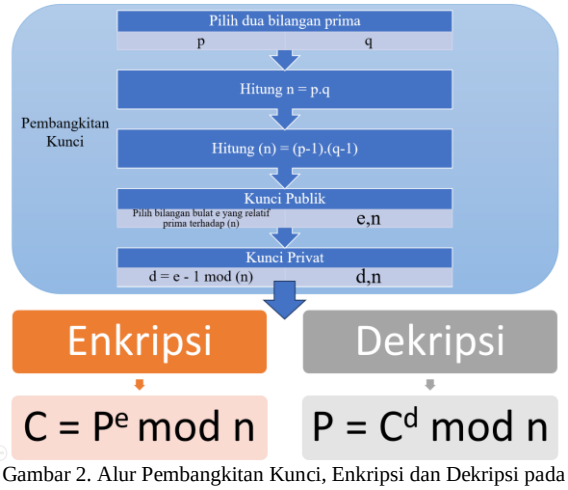
tanda tangan mereka [20]. Tanda tangan digital juga dapat diartikan sebagai tanda tangan elektronik yang digunakan untuk membuktikan keaslian identitas pengirim dari suatu pesan atau dokumen [21].

Di dalam kriptografi, tanda tangan digital adalah suatu nilai yang bergantung pada isi pesan dan pengirim pesan. Artinya, pesan yang memiliki isi berbeda meskipun dari pengirim yang sama, maka akan menghasilkan tanda tangan digital yang berbeda pula [22]. Salah satu contoh pemanfaatan tanda tangan digital adalah untuk legalisasi dokumen elektronik [23] seperti ijazah, sertifikat, surat, dan lain-lain.

Skema tanda tangan digital dibagi menjadi dua proses utama, yaitu penandatanganan (*sign*) dan verifikasi tanda tangan digital (*verify*) [24]. Proses penandatanganan dimulai dengan melakukan proses *hashing*, yang menghasilkan *message digest* (md_0), selanjutnya masuk ke dalam proses enkripsi dengan menggunakan kunci publik dan dihasilkan kode tanda tangan digital[25].

Sedangkan proses verifikasi dimulai dengan melakukan dekripsi pada kode tanda tangan digital untuk mengembalikan message digest asli (md_0), dilanjutkan memeriksa keaslian dokumen, dimana data pada dokumen yang diperiksa mausk proses *hashing* untuk menghasilkan *message digest* (md_1), lalu md_1 dibandingkan dengan md_0 [26]. Jika md_1 sama dengan md_0 maka dapat disimpulkan bahwa integritas data dalam dokumen tersebut terjamin, namun, jika sebaliknya, maka perlu dicurigai kemungkinan adanya modifikasi atau manipulasi data dalam dokumen tersebut[27].

2.2. Algoritma RSA

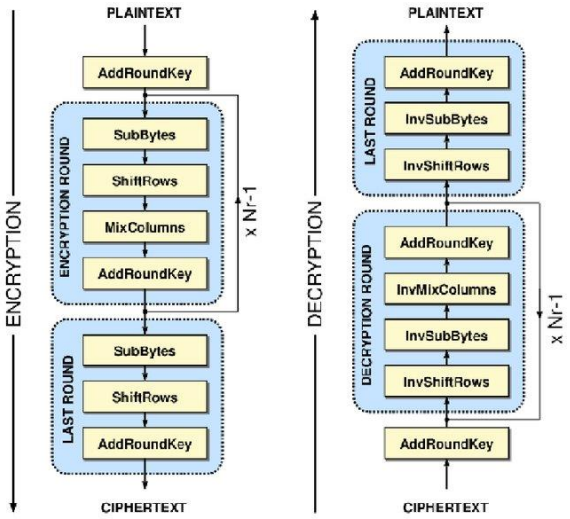


Gambar 2. Alur Pembangkitan Kunci, Enkripsi dan Dekripsi pada RSA

RSA merupakan algoritma kriptografi kunci asimetris, artinya mempunyai dua kunci yang berbeda yaitu kunci publik untuk proses enkripsi dan kunci privat untuk dekripsinya[28]. Algoritma RSA menggunakan pemfaktoran bilangan yang besar menjadi faktor-faktor prima, sehingga dapat ditemukan kunci privatnya[29]. Algoritma RSA memerlukan pembangkitan kunci yang besar untuk memiliki tingkat keamanan yang tinggi, dimana proses pembangkitan kunci tersebut dapat dilihat pada Gambar 2.

2.3. Algortima AES

AES adalah algoritma kriptografi sistem blok dengan kunci simetris, yang berarti bahwa kunci yang digunakan untuk mengenkripsi pesan adalah sama dengan kunci yang digunakan untuk mendekripsi pesan tersebut[30].



Gambar 3. Alur Enkripsi dan Dekripsi pada AES [32].

Seperti yang terlihat pada gambar 3, bahwa AES memiliki tahapan-tahapan utama yaitu[31]:

a. *Key Expansion*, yaitu perluasan kunci kunci asli yang memiliki panjang 128 bit diperluas menjadi serangkaian subkunci, yang akan digunakan dalam setiap putaran enkripsi.
b. *Initial Round*, yaitu putaran awal dimana plaintext di XOR dengan kunci putaran pertama.
c. *SubBytes* dimana setiap byte dari state (keadaan) digantikan dengan byte yang sesuai dari tabel substitusi S-box.
d. *ShiftRows*, yaitu bytes dalam state diubah berdasarkan pola tertentu.
e. *MixColumns*, yaitu kolom-kolom dalam state diubah menggunakan operasi matriks.
f. *AddRoundKey*, dimana state di XOR dengan kunci putaran yang sesuai. Setiap kolom dalam state di XOR dengan subkunci yang sesuai dari ekspansi kunci.
g. *Last Round* adalah putaran terakhir, yang terdiri SubBytes, ShiftRows, dan AddRoundKey seperti dalam putaran utama, namun tidak termasuk tahap MixColumns.

2.4. Metode *Hashing* MD5

Fungsi *hash* merupakan metode yang digunakan memampatkan pesan menjadi *message digest* (pesan ringkas) yang merupakan karakter yang merepresentasikan pesan aslinya, umumnya

berukuran jauh lebih kecil daripada ukuran pesan semula[33]. Fungsi *hash* MD5 (*Message-Digest Algorithm* 5) mampu menerima masukan berupa pesan berukuran sembarang dan menghasilkan keluaran dengan panjang 32 karakter atau 128 bit [19].

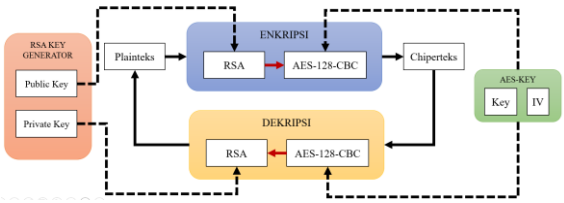
2.5. Nilai Entropi

Nilai Entropi mengacu pada ukuran seberapa acak atau tidak terduga suatu pesan atau informasi[34][35]. Semakin tinggi nilai entropi, semakin tidak terduga atau acak pesan tersebut, sedangkan nilai entropi yang rendah menunjukkan bahwa pesan tersebut lebih dapat diprediksi[36]. Nilai entropi sering diukur dalam satuan bit [37] dan digunakan dalam konteks kriptografi, kompresi data, dan teori informasi untuk mengukur seberapa baik pesan atau sumber informasi tertentu dapat dikompres atau dienkripsi.

3. HASIL DAN PEMBAHASAN

3.1. Perancangan Algoritma Super Enkripsi

Untuk meningkatkan keamanan data pada tanda tangan digital, maka penelitian ini merancang skema super enkripsi RSA dan AES. Proses enkripsi pada skema yang dirancang seperti pada gambar 4, dimana proses dimulai dengan enkripsi menggunakan RSA dan *public key*-nya, kemudian dilanjutkan enkripsi menggunakan algoritma AES dengan blok 128 bit dan mode operasi CBC (AES-128-CBC). Sedangkan untuk mengembalikan chiperteks menjadi plainteksnya, proses dimulai dengan enkripsi AES-128-CBC, kemudian dekripsi dilanjutkan menggunakan RSA dan *private key*-nya.



Gambar 4. Skema Super Enkripsi dan Dekripsi menggunakan RSA dan AES

Untuk pembangkitan kunci pada RSA, digunakan ukuran *private key* 512 bit, sedangkan untuk *initial vector* (IV) dan *key* pada AES digunakan ukuran masing-masing 16 bit. Ukuran kunci ini dipilih berdasarkan uji coba yang menekankan pada waktu proses dan ukuran file hasilnya.

Dalam penelitian ini, dokumen yang akan dilindungi datanya yaitu dokumen pengesahan laporan skripsi mahasiswa. Plainteks yang dimasukkan merupakan data yang tercantum pada lembar pengesahan skripsi, termasuk Judul Penelitian, Nama Mahasiswa, NIM, Dosen Pembimbing (Nama & NIDN), dan Tanggal Pengesahan. Contoh hasil super enkripsi disajikan dalam bentuk tabel, sebagaimana ditunjukkan pada Tabel 1. Plainteks ini sebelum masuk proses enkripsi diambil dulu *message digest* (*md*) sesuai pada skema tanda tangan digital, menggunakan MD5 sehingga menghasilkan ukuran md yang cukup kecil 128 bit. *Message digest* tersebut masuk proses enkripsi RSA menggunakan kunci publik kemudian chiperteks hasil RSA dienkripsi kembali menggunakan AES-128-CBC sehingga menghasilkan chiperteks hasil super enkripsi.

Tabel 1. Hasil Perancangan Super Enkripsi

Jenis	Data
Plainteks	Implementasi Tanda Tangan Digital Berbasis *QR-Code*|Diva Nuratnika Rahayu|101010|Fitri Nuraeni|999999|07/302023
Message Digest (md)	5ced6eaec585543a8ac00d4767d6c8cc (16 bytes)
Chiperteks RSA	PCtGleOeYus2sf+oIKt2Lw7ypOG+zhw2ivU/ UNgpaAi+xxWzZEPAJU9U/VBvu3DTcD7E Cx8RLx+fYtmksuComg (144 bytes)
Chiperteks SE	cE1jWFg0eGhBZ24zSWZIQmF3UHNCc1lB UGZPWVJWMTVxV3RsVGdJV0U1UW9tW EdkQ0grZlJnWFdqcTFjbXZiOUdRTDFzYytq VERFbmdMWUhjVlpXTXZBcTZ3cko5czlvb XZXbUVNc2JmMGc9 (144 bytes)
Hasil Dekripsi	5ced6eaec585543a8ac00d4767d6c8cc (16 bytes)

Sesuai tabel 1, untuk tahapan dekripsi dimulai dari chiperteks diproses menggunakan AES-128-CBC, kemudian dilanjutkan menggunakan algoritma RSA dengan *private key*-nya. Hasil ujicoba skema super enkripsi ini telah berhasil merubah *message digest* pesan menjadi chiperteks yang jauh berbeda, serta dapat mengembalikan *message digest* dari bentuk chiperteksnya.

3.2. Pengujian Algoritma Super Enkripsi

Rancangan super enkripsi yang sudah dibuat kemudian diuji untuk mengetahui lama proses dan tingkat keacakan chiperteks menggunakan nilai entropi. Pada uji coba ini digunakan 19 file teks yang dijadikan plainteks dengan ukuran yang berbeda mulai dari 136 bit sampai 1.336 bit. Pengujian yang dilakukan meliputi perbandingan waktu proses enkripsi dan dekripsi, serta nilai entropi dari setiap plainteks. Pengujian menggunakan aplikasi berbahasa php yang dikembangkan oleh penulis sendiri.

Tabel 2. Hasil Pengujian pada Algoritma Super Enkripsi

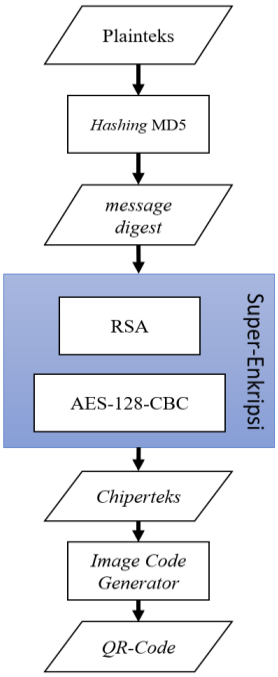
File ke-	Ukuran Plainteks (bytes)	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)	Nilai Entropi
1	136	0.298	0.516	5.997
2	200	0.163	0.151	6.063
3	268	0.134	0.150	5.988
4	336	0.129	0.150	6.017
5	400	0.123	0.156	6.088
6	468	0.145	0.131	6.038
7	536	0.117	0.156	6.163
8	600	0.133	0.158	6.013
9	668	0.237	0.217	6.003
10	736	0.172	0.163	6.088
11	800	0.175	0.133	6.053

12	868	0.126	0.186	6.013
13	936	0.175	0.177	6.047
14	1000	0.142	0.186	5.997
15	1068	0.221	0.142	5.997
16	1136	0.128	0.147	6.088
17	1200	0.134	0.169	5.972
18	1268	0.122	0.152	5.997
19	1336	0.128	0.150	6.013

Hasil pengujian dapat dilihat pada tabel 2, dimana rata-rata waktu proses enkripsi dari 0.158 *milisecond* (ms); rata-rata waktu proses dekripsi 0.178 ms; dan rata-rata nilai entropi dari chiperteks yang dihasilkan adalah 6.033 bit.

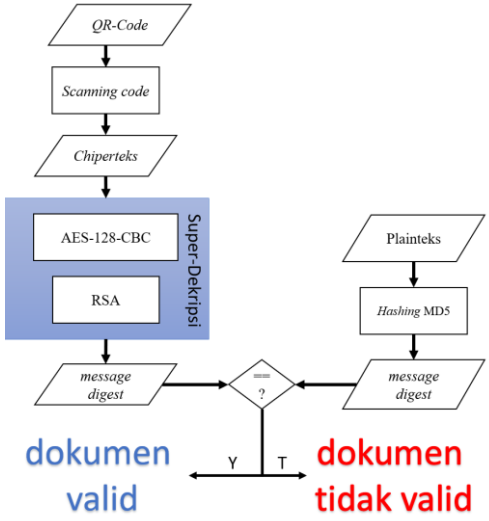
3.3. Perancangan Aplikasi Tanda Tangan Digital

Skema tanda tangan digital pada bahasan kriptografi mencakup dua langkah pokok, yakni langkah "*sign*" dan "*verify*". Oleh karen itu pada penelitian ini digunakan skema yang sama sebagaimana ditunjukkan dalam gambar 2 dan 3.



Gambar 5. Alur Proses *Sign*

Pada gambar 5, proses tanda tangan digital (*sign*) pada aplikasi yang dibangun diawali dengan memasukkan plainteks terlebih dahulu melalui proses *hashing* menggunakan fungsi MD5 sehingga menghasilkan (md_0). Selanjutnya, *message digest* (md_0) akan diarahkan ke dalam proses super enkripsi dengan menggunakan algoritma RSA dan AES-128. Proses enkripsi pertama menggunakan kunci publik dari algoritma RSA dan menghasilkan cipherteks. Cipherteks tersebut kemudian melewati proses enkripsi kedua dengan menerapkan algoritma AES berukuran blok 128 bit, dan menghasilkan cipherteks yang baru. Proses enkripsi ini menghasilkan kode tanda tangan digital yang akan digunakan dalam pembuatan *QR-Code* untuk memudahkan penyisipannya pada dokumen yang dituju.

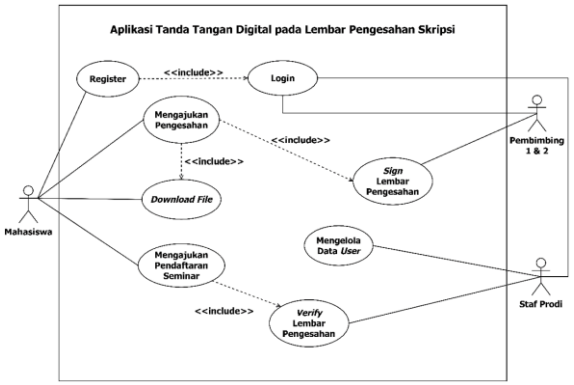


Gambar 6. Alur Proses *Verify*

Pada gambar 6, proses verifikasi dimulai dengan pemindaian kode *QR* untuk mendapatkan tanda tangan digital. Kode ini kemudian melewati dua tahap dekripsi, pertama melalui dekripsi menggunakan AES-128, dan selanjutnya melalui dekripsi menggunakan kunci pribadi pada algoritma RSA. Proses dekripsi ini akan menghasilkan pesan teracak asli (md_0) yang dibuat selama proses penandatanganan.

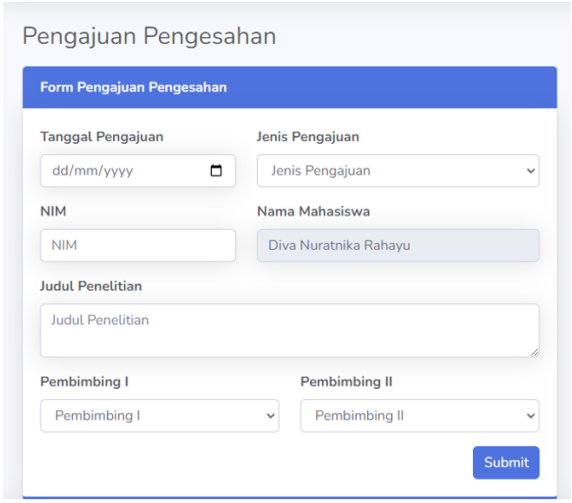
Dalam rangka memeriksa keatentikan dokumen, perlu ada teks asli dari dokumen tersebut. Data tersebut akan diarahkan ke dalam proses *hashing* untuk menghasilkan *message digest*/intisari pesan (md_1). Jika intisari pesan yang dihasilkan dari dokumen (md_1) sama dengan intisari pesan asli (md_0), maka keaslian data terjamin (integritas data terjaga). Namun, jika kedua intisari pesan berbeda, ini mengindikasikan bahwa dokumen tersebut palsu.

Perancangan aplikasi tanda tangan digital berbasis web dilakukan berdasarkan metode RUP. Dimana fase pertama yaitu pengidentifikasian proses bisnis, perumusan spesifikasi sistem, serta identifikasi aktor yang terlibat. Pada fase pertama ini, diketahui aktor yang melakukan pengesahan dokumen (*sign*) dan aktor yang melakukan pengecekan terhadap dokumen yang sudah disahkan menggunakan tanda tangan digital (*verify*).



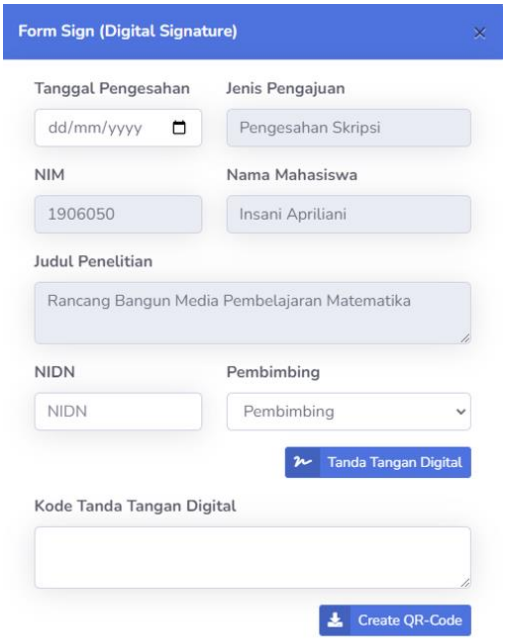
Gambar 7. *Use Case Diagram* Implementasi Tanda Tangan Digital

Fase kedua, mulai dirancangan skenario implementasi skema tanda tangan digital yang dibahas sebelumnya pada sebuah aplikasi pengesahan dokumen, dalam kasus ini yaitu dokumen pengesahan laporan skripsi. Pada skenario gambar 7, proses diawali dengan aktor mahasiswa mengajukan pengesahan dokumen skripsi melalui aplikasi. Proses pengesahan melalui *usecase sign* lembar pengesahan, dimana data dokumen oleh aktor Dosen, diproses menggunakan skema tanda tangan digital dengan super enkripsi RSA dan AES-128. Selanjutnya proses *verify* dilakukan oleh Aktor Staf Prodi yang menerima dokumen pengesahan sebagai syarat proses akademik.



Gambar 8. Antarmuka Pengajuan Pengesahan Dokumen

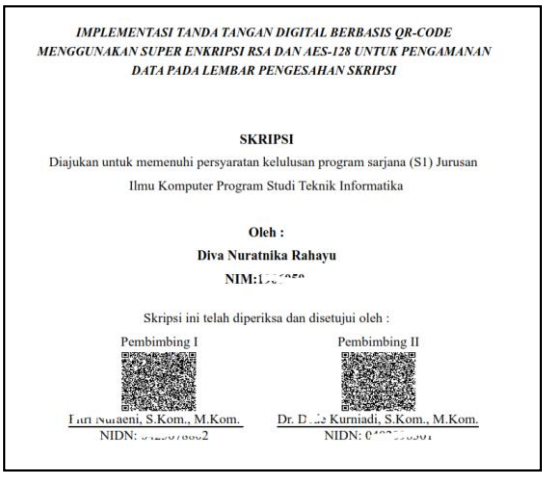
Pada fase ketiga, rancangan aplikasi ditranslate menggunakan bahasa pemrograman HTML dan PHP menjadi sebuah aplikasi berbasis website. Gambar 8 diatas merupakan antarmuka untuk aktor mahasiswa mengajukan dokumen skripsi untuk disahkan.



Gambar 9. Antarmuka Proses Pengesahan Dokumen

Antarmuka untuk proses *sign* dokumen menggunakan form pada Gambar 9. Proses super enkripsi terjadi saat tombol ”Tanda Tangan Digital” diklik, dan hasil dari proses ini berupa file gambar QR-*code* tanda tangan dari Aktor Pembimbing untuk dokumen pengesahan skripsi yang diajukan. Hasil proses *sign* ini dapat dilihat pada gambar 10 dimana *qr-code* tanda tangan digital sudah disisipkan pada dokumen pengesahan sebagai ganti tanda tangan manual.

Sedangkan untuk *verify* digunakan antarmuka halaman web, dimana aktor staf prodi akan mendekripsi kode tanda tangan digital hasil dari scan qr-code pada lembar pengesahan skripsi mahasiswa. Kemudian aktor staf prodi akan memverifikasi tanda tangan digital pada lembar pengesahan skripsi mahasiswa. Jika hasil dekripsi tanda tangan digital sama dengan hasil message digest dari lembar pengesahan, maka dapat dipastika bahwa draft skripsi tersebut asli. Namun, jika hasilnya berbeda, maka patut dicurigai adanya pemalsuan.



Gambar 10. File Hasil Pengesahan Dokumen

Fase selanjutnya dalam perancangan aplikasi tanda tangan digital ini adalah pengujian fitur-fitur aplikasi menggunakan metode *blackbox*. Tabel 3 memperlihatkan hasil pengujian aplikasi, dimana seluruh fitur yang dirancang sesuai kebutuhan proses bisnis pengesahan dokumen telah berjalan sesuai skenario.

Tabel 3. Hasil Pengujian Aplikasi Tanda Tangan Digital

Aktivitas	Kelas Uji	Skenario Uji	Hasil
Pengajuan	Mengajukan pengesahan proposal atau skripsi	Mengisi tanggal pengajuan, jenis pengajuan, nim, nama. Judul, p1 dan p2, lalu menekan tombol submit	Valid
	Melihat riwayat pengajuan	Melihat status pengajuan, untuk memeriksa jika lembar pengesahan telah ditandatangani	Valid
Sign	Membuat tanda tangan digital	Menekan tombol tanda tangan digital pada *form sign* yang berisi data lembar	Valid

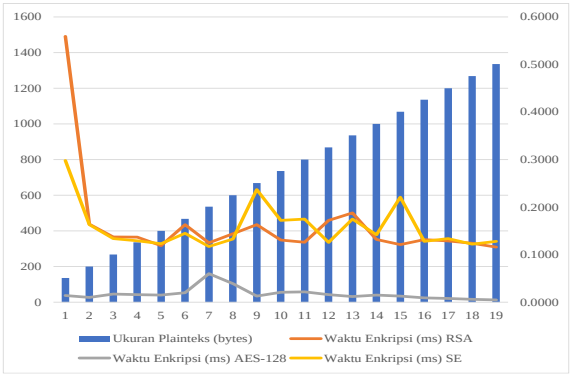
Aktivitas	Kelas Uji	Skenario Uji	Hasil
		pengesahan skripsi mahasiswa	
	Membuat *qr-code*	Menekan tombol *create qr-code* untuk menyisipkan kode tanda tangan digital	Valid
	Melihat riwayat pengesahan	Melihat status pengesahan, untuk membuktikan bahwa status telah berubah menjadi *signed*	Valid
Verifikasi	Melakukan dekripsi pada kode tanda tangan digital	Memasukkan hasil *scan qr-code* yang berisi kode tanda tangan digital, kemudian menekan tombol dekripsi	Valid
	Melakukan *hashing* dari data lembar pengesahan	Memasukkan data-data yang tertera pada lembar pengesahan, kemudian menekan tombol verifikasi	Valid
	Proses verifikasi tanda tangan digital	Mencocokkan hasil dekripsi kode tanda tangan digital dan hasil *hash* lembar pengesahan	Valid

4. DISKUSI

Pada penelitian ini rumusan masalah yang pertama dibahas adalah bagaimana merancangan skema tanda tangan digital yang menerapkan super enkripsi RSA dan AES-128 bit. Pada tabel 1 sudah terlihat bagaimana plainteks diproses menjadi kode enkripsi (chiperteks). Proses perancangan tanda tangan digital diawali dengan mengambil instisari (*message digest*) dari data lembar pengesahan skripsi menggunakan fungsi hash MD5. Hasil message digest tersebut akan melalui proses super enkripsi menggunakan algoritma RSA dan AES - 128 sehingga menghasilkan kode tanda tangan digital. Selanjutnya, kode tanda tangan digital akan dikonversi ke dalam QR-Code untuk memudahkan proses penyisipannya pada lembar pengesahan skripsi [38], dan hal ini sesuai dengan hasil penelitian yang dilakukan pada tahun 2019.

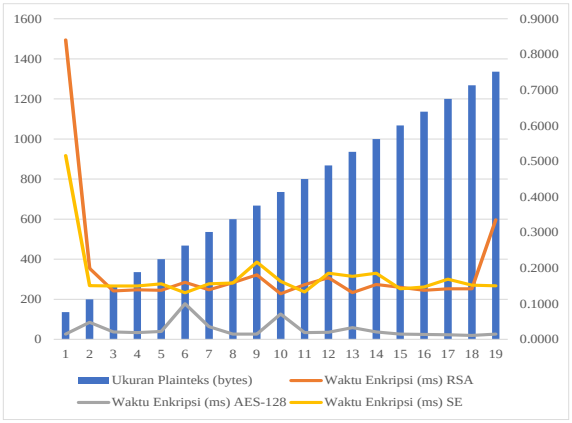
Adapun proses verifikasi tanda tangan digital diawali dengan membaca QR-Code terlebih dahulu untuk mendapatkan kode tanda tangan digital. Kemudian kode tersebut masuk proses dekripsi ganda dengan AES - 128 dan RSA. Proses dekripsi ini akan mengembalikan message digest asli yang dibuat pada proses tanda tangan. Sedangkan untuk memeriksa keaslian dokumen skripsi mahasiswa, diperlukan adanya inputan data dari lembar pengesahan. Data tersebut masuk proses *hashing* untuk mendapatkan *message digest*. Jika *message digest* yang didapat dari lembar pengesahan ini sama dengan *message digest* asli, maka dapat dipastikan dokumen tersebut asli. Namun jika dihasilkan *message digest* yang berbeda, maka patut dicurigai adanya modifikasi dan fabrikasi data pada dokumen tersebut.

Selain pengujian pada algoritma super-enkripsi, hasil penelitian ini juga dibandingkan dengan hasil enkripsi yang menggunakan algoritma RSA. Algoritma RSA dipilih untuk dibandingkan karena pada skema tanda tangan digital yang digunakan adalah algoritma kriptografi asimetris[39], salah satunya yaitu RSA.



Gambar 11. Perbandingan waktu enkripsi algoritma RSA, AES dan super enkripsi RSA & AES-128

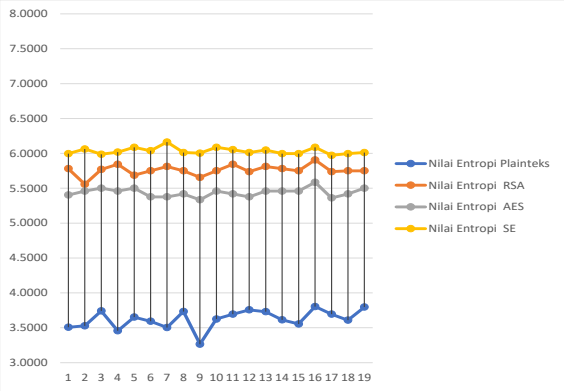
Pada Gambar 11 dapat dilihat bahwa ukuran plainteks tidak mempengaruhi waktu enkripsi karena pada proses digital signature ini digunakan *message digest* (intisari pesan) hasil dari proses *hashing* MD5. Pesan yang diproses menggunakan MD5 akan menghasilkan intisari pesan dalam ukuran 128 bit[40]. Waktu yang dibutuhkan untuk enkripsi 20 file tersebut dapat dikatakan fluktuatif hal ini dipengaruhi oleh ukuran file kunci publik dan kunci private RSA yang panjang [41]. Walau demikian berdasarkan grafik tersebut, waktu proses super enkripsi masih tidak jauh berbeda dengan waktu enkripsi RSA. Rata-rata waktu enkripsi RSA adalah 0.1622 milisecond dan super enkripsi RSA+AES-128-CBC adalah 0.1580 milisecond. Hal ini karena waktu proses AES-128-CBC yang ditambahkan pada skema RSA memiliki waktu enkripsi yang relatif singkat[42], sehingga tidak menambah waktu proses menjadi lebih lama.



Gambar 12. Perbandingan waktu dekripsi algoritma RSA, AES dan super enkripsi RSA & AES-128

Pada Gambar 12 dapat dilihat bahwa waktu yang dibutuhkan untuk dekripsi 20 file tersebut dapat dikatakan cukup singkat, dimana waktu dekripsi algoritma super enkripsi relatif sama dengan waktu

dekripsi RSA saja. Rata-rata waktu dekripsi untuk RSA adalah 0.1966 milisecond sedangkan untuk algoritma super enkripsi adalah 0.1784 milisecond. Waktu dekripsi lebih lama dibanding waktu enkripsi hal ini dikarenakan proses dekripsi RSA yang cukup lama dipengaruhi oleh kunci private yang cukup besar[43].



Gambar 13. Perbandingan nilai entropi plainteks dan chiperteks hasil RSA, AES dan super enkripsi RSA & AES-128

Pada Gambar 13, nilai rata-rata entropi menggunakan super enkripsi RSA dan AES-128 adalah 6.0332, sedangkan untuk nilai rata-rata entropi menggunakan AES-128 adalah 5.4386 dan RSA adalah 5.7594. Dengan perbedaan 0.2738, algoritma super enkripsi menghasilkan nilai entropi yang lebih ideal karena mendekati angka 8 [44], sehingga hasil enkripsi ini cukup sulit untuk dianalisis atau terkena serangan oleh pihak yang tidak berhak[45].

5. KESIMPULAN

Berdasarkan hasil penelitian ini penggunaan super enkripsi RSA dan AES telah terbukti memiliki kinerja yang lebih baik dalam pengamanan data. Hal ini terbukti dengan waktu proses enkripsi yang lebih cepat, dimana rata-rata waktu enkripsi RSA adalah 0.1622 milisecond sedangkan super enkripsi RSA+AES-128-CBC adalah 0.1580 milisecond. Kemudian untuk waktu proses dekripsi relatif mendekati waktu enkripsi RSA, dimana rata-rata waktu dekripsi untuk RSA adalah 0.1966 milisecond sedangkan untuk algoritma super enkripsi adalah 0.1784 milisecond. Terakhir perbandingan nilai entropi untuk chiperteks hasil super enkripsi lebih baik dibanding RSA dan AES-128, dengan nilai yang lebih mendekati 8. Sehingga, algoritma super enkripsi RSA dan AES-128-CBC ini layak untuk skema pengamanan pada tanda tangan digital berbasis QR-code.

Untuk pengembangan penelitian yang serupa dikemudian hari dapat dilakukan perbaikan pada perancangan aplikasi dengan alur proses *sign* dan *verify* yang lebih sederhana. Selain itu, dapat dilakukan perbaikan skema tanda tangan digital yang mampu menghasilan ukuran chiperteks yang lebih kecil.

DAFTAR PUSTAKA

[1] A. Al Omar *et al.*, "A Transparent and Privacy-Preserving Healthcare Platform With Novel Smart Contract for Smart Cities," *IEEE Access*, vol. 9, pp. 90738–90749, 2021, doi: 10.1109/ACCESS.2021.3089601.

[2] M. Marsaid, R. H. Jan, M. Huda, E. L. Lydia, and SHankarK, "IMPORTANCE OF DATA SECURITY IN BUSINESS MANAGEMENT PROTECTION OF COMPANY AGAINST SECURITY THREATS," *J. Crit. Rev.*, vol. 7, no. 01, Jan. 2020, doi: 10.31838/jcr.07.01.45.

[3] E. Karaarslan and M. Babiker, "Digital Twin Security Threats and Countermeasures: An Introduction," in *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*, Dec. 2021, pp. 7–11, doi: 10.1109/ISCTURKEY53027.2021.9654360.

[4] C. Alcaraz and J. Lopez, "Digital Twin: A Comprehensive Survey of Security Threats," *IEEE Commun. Surv. Tutorials*, vol. 24, no. 3, pp. 1475–1503, 2022, doi: 10.1109/COMST.2022.3171465.

[5] H. M. Alzoubi *et al.*, "Cyber Security Threats on Digital Banking," in *2022 1st International Conference on AI in Cybersecurity (ICAIC)*, May 2022, pp. 1–4, doi: 10.1109/ICAIC53980.2022.9896966.

[6] S. Samsoni *et al.*, "Implementasi Sistem Keamanan Komputer Host Menggunakan Sistem Operasi Fedora Linux," *Innov. J. Soc. Sci. Res.*, vol. 3, no. 2, pp. 721–736, 2023.

[7] D. Ingle, M. Kulkarni, P. Shinde, and M. Tambe, "Literature Review of Data Security Measures and Access Control Mechanisms of Information Security," 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:248425917.

[8] A. Ali, B. A. S. Al-rimy, F. S. Alsubaei, A. A. Almazroi, and A. A. Almazroi, "HealthLock: Blockchain-Based Privacy Preservation Using Homomorphic Encryption in Internet of Things Healthcare Applications," *Sensors*, vol. 23, no. 15, p. 6762, Jul. 2023, doi: 10.3390/s23156762.

[9] F. N. Hasan, "Implementasi Sistem Business Intelligence Untuk Data Penelitian di Perguruan Tinggi," in *Prosiding Seminar Nasional Teknoka*, Nov. 2019, vol. 4, pp. I1–I10, doi: 10.22236/teknoka.v4i1.3943.

[10] A. Ayub Khan, A. A. Laghari, A. A. Shaikh, S. Bourouis, A. M. Mamlouk, and H. Alshazly, "Educational Blockchain: A Secure Degree Attestation and Verification Traceability Architecture for Higher

Education Commission," *Appl. Sci.*, vol. 11, no. 22, p. 10917, Nov. 2021, doi: 10.3390/app112210917.

[11] T. Wellem, Y. Nataliani, and A. Iriani, "Academic Document Authentication using Elliptic Curve Digital Signature Algorithm and QR Code," *JOIV Int. J. Informatics Vis.*, vol. 6, no. 3, p. 667, Sep. 2022, doi: 10.30630/joiv.6.2.872.

[12] W. van Donge, N. Bharosa, and M. F. W. H. A. Janssen, "Data-driven government: Cross-case comparison of data stewardship in data ecosystems," *Gov. Inf. Q.*, vol. 39, no. 2, p. 101642, Apr. 2022, doi: 10.1016/j.giq.2021.101642.

[13] S. R. Moulick, "Review of: Digital Signatures by Jonathan Katz," *ACM SIGACT News*, vol. 46, no. 1, pp. 10–12, Mar. 2015, doi: 10.1145/2744447.2744450.

[14] F. Nuraeni, S. Tinggi, T. Garut, and D. Kurniadi, "Implementasi Skema QR-Code dan Digital Signature menggunakan Kombinasi Algoritma RSA dan AES untuk Pengamanan Data Sertifikat Elektronik Academic Information System View project Educational Data Mining View project," pp. 43–52, 2020, [Online]. Available: https://www.researchgate.net/publication/349277434.

[15] A. Saepulrohman and A. Ismangil, "Data integrity and security of digital signatures on electronic systems using the digital signature algorithm (DSA)," *Int. J. Electron. Commun. Syst.*, vol. 1, no. 1, pp. 11–15, Jun. 2021, doi: 10.24042/ijecs.v1i1.7923.

[16] B. Harjito, T. Setyawati, and A. Wijayanto, "Comparative Analysis between Elgamal and NTRU Algorithms and their implementation of Digital Signature for Electronic Certificat," *Int. J. Electr. Comput. Eng. Syst.*, vol. 13, no. 9, pp. 729–739, Dec. 2022, doi: 10.32985/ijeces.13.9.1.

[17] W. Ariandi, S. Widyastuti, and L. Haris, "Implementasi Block Cipher Electronic Codebook (ECB) untuk Pengamanan Data Pegawai," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 2, no. 02, pp. 65–74, 2020, doi: 10.46772/intech.v2i02.291.

[18] F. A. E. F. Faris, F. Y. Febi, I. I. Iwan, and P. Pizaini, "Kombinasi algoritma kriptografi vigenere cipher dengan metode zig-zag dalam pengamanan pesan teks," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 4, no. 1, pp. 182–192, Apr. 2023, doi: 10.37859/coscitech.v4i1.4787.

[19] H. Mursid, J. Supardi, and M. Q. Rizkie, "Pengujian Integritas File Operasi Tanda Tangan Digital Menggunakan Kombinasi Hash MD5, RSA dan Skema Qr-Cod," *Generic*, vol. 14, no. 2, pp. 30–37, 2022.

[20] Y. Anshori, A. Y. Erwin Dodu, and D. M. P. Wedananta, "Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital," *Techno.Com*, vol. 18, no. 2, pp. 110–121, 2019, doi: 10.33633/tc.v18i2.2166.

[21] Y. Fitriyah, "Analisis Tingkat Kesiapan implmentasi Tanda Tangan Digital Untuk Autentikasi Dokumen Rekam Medis ELektronik di Instalasi Rawat Jalan RSUD Kota Yogyakarta," *J. Inf. Syst. Public Heal.*, vol. 7, no. 2, p. 53, Aug. 2022, doi: 10.22146/jisph.73666.

[22] R. Munir, *Kriptografi 2nd Edition*, 2nd ed. Bandung: Informatika, 2019.

[23] R. Dermawan, "Pemanfaatan Tanda Tangan Digital Tersertifikasi di Era Pandemi," *J. Huk. Lex Gen.*, vol. 2, no. 8, pp. 762–781, Aug. 2021, doi: 10.56370/jhlg.v2i8.95.

[24] A. Saepulrohman and T. P. Negara, "Implementasi Algoritma Tanda Tangan Digital Berbasis Kriptografi Kurva Eliptik Diffie-Hellman," *Komputasi J. Ilm. Ilmu Komput. dan Mat.*, vol. 18, no. 1, pp. 22–28, 2021, doi: 10.33751/komputasi.v18i1.2569.

[25] T. Abdurrachman and B. R. Suteja, "Pengembangan Sistem Informasi Asosiasi Jasa Konstruksi dengan Menerapkan Tanda Tangan Digital," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 261–273, 2021, doi: 10.28932/jutisi.v7i1.3431.

[26] W. Sholihah, S. Indriasari, I. Noviyanti, A. Mardiyono, and N. Aziezah, "ESVISIGN: Tanda Tangan Digital Sekolah Vokasi IPB," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 3, no. 4, pp. 217–226, 2022, doi: 10.35746/jtim.v3i4.188.

[27] A. Nadzifarin and A. Asmunin, "Penerapan Elliptic Curve Digital Signature Algorithm pada Tanda Tangan Digital dengan Studi Kasus Dokumen Surat – Menyurat," *J. Informatics Comput. Sci.*, vol. 4, no. 01, pp. 1–9, 2022, doi: 10.26740/jinacs.v4n01.p1-9.

[28] N. B. N. Putra, F. A. Raihana, W. M. A. Mondong, and A. R. Kardian, "Analisis Enkripsi Kriptografi Asimetris Algoritma RSA Berbasis Pemrograman Batch pada Media Flashdisk," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, no. 1, pp. 142–154, 2023, [Online]. Available: https://tunasbangsa.ac.id/ejurnal/index.php/jurasik.

[29] M. Rizki and F. Ariyani, "PENERAPAN KRIPTOGRAFI DENGAN MENGGUNAKAN ALGORITMA RSA UNTUK PENGAMANAN DATA

BERBASIS DESKTOP PADA PT TRIAS MITRA JAYA MANUNGGAL," *Sist. Komput. dan Tek. Inform.*, vol. 4, no. 2, pp. 1–6, 2021.

[30] N. Cristy and F. Riandari, "Implementasi Metode Advanced Encryption Standard (AES 128 Bit) untuk Mengamankan Data Keuangan," *JIKOMSI [Jurnal Ilmu Komput. dan Sist. Informasi]*, vol. 4, no. 2, pp. 75–85, 2021, [Online]. Available: https://ejournal.sisfokomtek.org/index.php/jikom/article/view/181%0A.

[31] I. A. R. Simbolon, I. Gunawan, I. O. Kirana, R. Dewi, and S. Solikhun, "Penerapan Algoritma AES 128-Bit dalam Pengamanan Data Kependudukan pada Dinas Dukcapil Kota Pematangsiantar," *J. Comput. Syst. Informatics*, vol. 1, no. 2, pp. 54–60, 2020.

[32] S. Arrag, "Design and Implementation A different Architectures of mixcolumn in FPGA," *Int. J. VLSI Des. Commun. Syst.*, vol. 3, no. 4, pp. 11–22, 2012, doi: 10.5121/vlsic.2012.3402.

[33] I. Rahim, N. Anwar, A. M. Widodo, K. Karsono Juman, and I. Setiawan, "Komparasi Fungsi Hash Md5 Dan Sha256 Dalam Keamanan Gambar Dan Teks," *Ikraith-Informatika*, vol. 7, no. 2, pp. 41–48, 2022, doi: 10.37817/ikraith-informatika.v7i2.2249.

[34] R. Rihartanto, R. K. Ningsih, A. F. O. Gaffar, and D. S. B. Utomo, "Implementation of vigenere cipher 128 and square rotation in securing text messages," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 3, pp. 201–209, 2020, doi: 10.14710/jtsiskom.2020.13476.

[35] N. Syafitri, S. Farradinna, W. Jayanti, and Y. Arta, "Machine Learning To Create Decision Tree Model To Predict Outcome of Enterpreneurship Psychological Readiness (Epr) Machine Learning Untuk Membuat Model Decision Tree Guna," *J. Tek. Inform.*, vol. 4, no. 2, pp. 381–390, 2023, [Online]. Available: https://www.jutif.if.unsoed.ac.id/index.php/jurnal/article/download/590/297.

[36] L. B. Handoko and C. Umam, "Kombinasi Vigenere-Aes 256 dan Fungsi Hash Dalam Kriptografi Aplikasi Chatting," *Pros. Sains Nas. dan Teknol.*, vol. 12, no. 1, p. 390, Nov. 2022, doi: 10.36499/psnst.v12i1.7068.

[37] F. Elfaladonna and A. Rahmadani, "Analisa Metode Classification-Decission Tree Dan Algoritma C.45 Untuk Memprediksi Penyakit Diabetes Dengan Menggunakan Aplikasi Rapid Miner," *SINTECH (Science Inf. Technol. J.*, vol. 2, no. 1, pp. 10–17, 2019, doi: 10.31598/sintechjournal.v2i1.293.

[38] 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 Sertifikat Elektronik," in *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, 2020, p. 43.

[39] M. Taufiqurrahman, Irawan, and I. Syamsuddin, "Perancangan Sistem Tanda Tangan Digital (Digital Signature)," in *Seminar Nasional Teknik Elektro dan Informatika*, 2020, pp. 60–65.

[40] G. P. Reddy, A. Narayana, P. K. Keerthan, B. Vineetha, and P. Honnavalli, "Multiple hashing using SHA-256 and MD5," in *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 1*, 2021, pp. 643–655.

[41] T. H. Saputro, N. H. Hidayati, and E. I. H. Ujianto, "Survei Tentang Algoritma Kriptografi Asimetris," *J. Inform. Polinema*, vol. 6, no. 2, pp. 67–72, 2020, doi: 10.33795/jip.v6i2.345.

[42] L. Laurentinus, H. A. Pradana, D. Y. Sylfania, and F. P. Juniawan, "Performance comparison of RSA and AES to SMS messages compression using Huffman algorithm," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 3, pp. 171–177, Jul. 2020, doi: 10.14710/jtsiskom.2020.13468.

[43] F. D. Yonathan, H. Nasution, and H. Priyanto, "Aplikasi Pengaman Dokumen Digital Menggunakan Algoritma Kriptografi Hybrid dan Algoritma Kompresi Huffman," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 2, p. 181, Aug. 2021, doi: 10.26418/jp.v7i2.47077.

[44] D. W. Utom and C. A. Sari, "Optimalisasi Vigenere dan Beaufort Cipher Menggunakan Teknik Fibonacci Untuk Citra Digital Optimization of Vigenere and Beaufort Ciphers Using Fibonacci Techniques for," in *Seminar Nasional Inovasi dan Pengembangan Teknologi Terapan (SENOVTEK)*, 2022, pp. 35–44.

[45] E. Supriyanto, W. T. Handoko, S. A. Wibowo*, and E. Ardhianto, "Peningkatan Ketahanan Algoritma Vigenere menggunakan Generator kunci Tiga Lapis," *J. MAHAJANA Inf.*, vol. 7, no. 1, pp. 24–33, Jun. 2022, doi: 10.51544/jurnalmi.v7i1.2894.