Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

Implementasi Secure Hash Algorithm-256 dan Advanced Encryption Standard Untuk Verifikasi Tanda Tangan Digital

Ahmad Zaky Nadimsyah, Muhammad Ezar Al Rivan*

Fakultas Ilmu Rekayasa dan Komputer, Informatika, Universitas Multi Data Palembang, Palembang
Jl. Rajawali No.14, 9 Ilir, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan, Indonesia
Email: [1]nadim@mhs.mdp.ac.id, [2,*]meedzhar@mdp.ac.id
Email Penulis Korespondensi: nadim@mhs.mdp.ac.id
Submitted: **21/09/2024;** Accepted: **09/10/2024;** Published: **15/10/2024**

Abstrak−Urusan surat menyurat dilakukan diseluruh lapisan masyarakat yang merupakan bagian dari negara hukum pada urusan pekerjaan atau kegiatan sehari hari yang menyangkut mengenai kewajiban seorang masyarakat secara hukum. Penandatanganan oleh semua pihak terkait wajib dilakukan dengan menandatangani dokumen dengan tanda tangan basah untuk mengikat maksud dan isi surat dengan semua pihak yang terkait sehingga memiliki nilai hukum. Tanda tangan basah mulai ditinggalkan seiring berkembangnya akses internet yang membuat peralihan infrastruktur kebutuhan surat menyurat mulai menjadi sistem hybrid dimana penanda tanganan dokumen bisa dilakukan secara elektronik. Hal ini membuat munculnya potensi pemalsuan tanda tangan elektronik karena tidak ada validasi yang bisa dilakukan terhadap suatu citra digital tanda tangan elektronik. Mudahnya proses pemindaian tanda tangan basah seseorang dapat menjadi potensi untuk di salah gunakan oleh pihak yang tidak berwenang sehingga dikembangkan lagi tanda tangan digital untuk mengatasi masalah ini. Tanda tangan digital digunakan untuk mengamankan pesan atau dokumen dari pihak yang tidak berhak atau berwenang, mengamankan data sensitif, menguatkan kepercayaan pihak yang menandatangan dan mendeteksi upaya perusakan. Solusi yang ditawarkan pada penelitian ini adalah pengembangan suatu sistem perangkat lunak yang mampu memverifikasi keaslian dokumen digital agar dokumen tidak disalahgunakan dan dapat digunakan sebagaimana mestinya menggunakan Fungsi Hash Secure Hash Algorithm (SHA)-256 dan Algoritma Enkripsi Advanced Encryption Standard (AES). Perangkat lunak yang dihasilkan memiliki persentase kepuasan sebesar 93,71 % , sehingga dapat disimpulkan bahwa aplikasi yang telah dikembangkan dapat berjalan dengan baik.

Kata Kunci: AES; Enkripsi; Dokumen Digital; Hash; SHA-256

Abstract−Process of traditional document approval carried out across all layers of society as part of a legal system in matters related to work or daily activities that involve legal obligations by all relevant parties is done by using handwritten signatures to bind the intentions and contents of the document with all related parties and provie legal value. With the development of internet access, traditional signatures have started to be abandoned, leading to a shift in document approval infrastructure towards a hybrid system where document signing can be done electronically. This shift has introduced the potential for electronic signature forgery, as there is no validation that can be performed on a digital image of an electronic signature.The ease of scanning someone's traditional signature can potentially be misused by unauthorized parties, so digital signatures have been developed to address this problem. Digital signatures are used to secure messages or documents from unauthorised parties, secure sensitive data, strengthen the signatories' confidence and detect attempted corruption. The solution offered in this project is the development of a software system capable of verifying the authenticity of digital documents so that the documents are not abused and can be used as appropriate using the SHA-256 Hash Function and the AES-256 Encryption Algorithm. The software produced has a satisfaction percentage of 93.71%, so it can be concluded that the applications that have been developed are running well.

Keywords: AES; Encryption; Digital Document; Hash; SHA-256

1. PENDAHULUAN

Urusan surat menyurat dilakukan diseluruh lapisan Masyarakat yang merupakan bagian dari negara Hukum pada urusan pekerjaan atau kegiatan sehari hari yang menyangkut mengenai kewajiban seorang Masyarakat secara hukum. Penandatanganan oleh semua pihak terkait wajib dilakukan dengan menandatangani dokumen dengan tanda tangan basah untuk mengikat maksud dan isi surat dengan semua pihak yang terkait sehingga memiliki nilai hukum [1]. Tanda tangan basah dalam sebuah dokumen surat sudah diakui keabsahannya di mata hukum [2].

Tanda tangan basah mulai ditinggalkan seiring berkembangnya akses internet yang membuat peralihan infrastruktur kebutuhan surat menyurat mulai menjadi sistem hybrid dimana dokumen fisik masih diperlukan namun penanda tanganan dokumen bisa dilakukan secara elektronik [3]. Hal ini membuat munculnya potensi pemalsuan tanda tangan elektornik karena tidak ada validasi yang bisa dilakukan terhadap suatu citra digital tanda tangan elektronik [4]. Kasus pemalsuan tanda tangan basah maupun elektronik mulai bermunculan di instansi civitas akademik yang menguntungkan oknum tertentu dan merugikan pihak lainnya. Pada tahun 2014 kasus pemalsuan tanda tangan terjadi di Program Pascasarjana suatu universitas [5], selain itu di tahun 2016 kasus pemalsuan identitas dan pemalsuan tanda tangan dilakukan oleh warga yang menolak pendirian pabrik PT Semen Indonesia di daerah Rembang [6]. Pada tahun 2017 dua orang dari Indonesian Enterpreneur Club (IEC) memalsukan stempel dan tanda tangan presiden badan eksekutif mahasiswa (BEM) di suatu Universitas [7].Pada tahun 2022 mahasiswa suatu Universitas di lampung melakukan pemalsuan tanda tangan rektor di universitas tersebut untuk hak uji materi peraturan perundang undangan [8] dan pada tahun 2023 pemalsuan tanda tangan Gubernur Kalimantan Timur dilakukan oleh inspektorat daerah pada kasus izin usaha pertambangan [9].

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

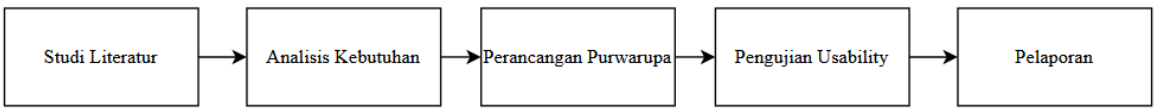
Penelitian oleh [10] membahas mengenai pemanfaatan tanda tangan digital untuk mendukung program Green Information and Communication Technology (Green ICT), dengan bertujuan salah satunya adalah untuk mengurangi penggunaan kertas di lingkungan perkantoran. Metode yang digunakan adalah Public Key Cryptographic Standard #12, karena metode ini tidak memerlukan infrastruktur tersendiri sehingga dapat lebih menghemat biaya. Solusi pendekatan pertama adalah Digital Signature System atau Dsign adalah sistem yang dikembangkan berbasis Software as a Service (SaaS), sistem ini mengimplementasi Fungsi SHA-1 untuk tanda tangan digital pada dokumen elektronik [11]. Solusi ini hanya menerapkan Fungsi Hash untuk integritas dokumen, namun tidak menerapkan enkripsi sama sekali [12]. Solusi pendekatan kedua adalah sistem yang dikembangkan oleh [13] melakukan penerapan tanda tangan digital menggunakan Quick Response Code (QR Code) dengan Algoritma Advanced Encryption Standard (AES) pada dokumen permintaan barang. Solusi ini hanya menerapkan enkripsi isi dokumen tanpa fungsi hash untuk menyembunyikan informasi pada dokumen tersebut. Solusi pendekatan ketiga adalah sistem yang dikembangkan oleh [14] melakukan penerapan Algoritma RSA pada tanda tangan digital. Solusi ini menerapkan fungsi hash dan enkripsi yang menggunakan Assymetric Key yang menghasilkan Public Key untuk kebutuhan enkripsi dan Private Key untuk kebutuhan dekripsi [15].

Pemilihan algoritma yang digunakan dilakukan berdasarkan batasan masalah dan memilih atau mengkombinasikan alternatif solusi dari penilitian terkait yang dijelaskan pada subbab ini. Pembuktian keaslian dokumen digital di civitas akademika membutuhkan agar sistem verifikasi dokumen digital bisa diakses oleh banyak orang, sehingga Algoritma RSA yang membutuhkan private Key untuk dekripsi ciphertext tidak cocok digunakan. Penelitian oleh [11]. hanya menggunakan fungsi Hash untuk pencocokan dokumen tetapi tidak mengimplementasikan metode Enkripsi apapun. Penelitian oleh [13] hanya menggunakan Algoritma AES untuk kebutuhan enkripsi tetapi tidak menggunakan fungsi Hash untuk isi informasi dokumen, sehingga alih alih membuktikan keaslian dokumen, proses dekripsi pada [13] akan mengekspos isi dokumen tersebut yang bisa diakses secara public. Solusi yang diajukan untuk pengembangan projek ini adalah gabungan dari semua alternatif solusi penelitian terkait yang telah dijelaskan, untuk mengurangi kelemahan masing masing alternatif solusi dan memperbaiki tingkat keamanan data serta mempermudah distribusi dokumen Digital. Solusi pada pengembangan projek ini akan menggunakan fungsi Hash SHA-256 untuk membuat message digest berdasarkan isi informasi penting dari dokumen, lalu mengenkripsi message digest tersebut menggunakan Algoritma AES dengan panjang key 256-bit dan memasukan nilai ciphertext yang dihasilkan ke dalam QR Code. Key yang digunakan untuk dekripsi dan enkripsi akan disimpan secara aman pada basis data dan nilai Key akan berbeda untuk setiap pihak yang memberi persetujuan pada dokumen digital.

Tujuan dari penelitian ini adalah mengembangkan perangkat lunak yang mampu menerbitkan dokumen digital dan secara aman mendistribusikannya serta mampu memverifikasi keaslian dokumen digital agar tidak disalah gunakan dan dapat digunakan sebagaimana mestinya.

2. METODOLOGI PENELITIAN

Pada penelitian ini terdapat 5 tahapan utama, yaitu studi literatur untuk menjadi dasar dari penelitian, dilanjutkan dengan analisis kebutuhan, perancangan purwarupa berdasarkan hasil analisis, pengujian usability kepada end user, dan diakhiri dengan pembuatan laporan. Tahapan-tahapan tersebut dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1 Studi Literatur

Tahapan pertama pada alur penelitian ini adalah studi literatur sebagai dasar landasan teoritis, serta pijakan penelitian agar lebih terarah dan kontekstual [16]. Tahapan ini berfungsi untuk mengidentifikasi teori dan temuan terbaru yang berkaitan dengan penelitian yang dilakukan. Di samping itu, studi literatur juga memudahkan pemahaman yang lebih mendalam tentang algoritma yang dibutuhkan untuk aplikasi yang dikembangkan. Dengan menggabungkan pengetahuan ini, diharapkan dapat mengarahkan penelitian menuju pengembangan solusi yang sesuai dengan requirement.

Adapun beberapa algoritma yang digunakan pada penelitian terkait adalah sebagai berikut :

a. Secure Hash Algorithm (SHA-1)
   Digital Signature System atau Dsign adalah sistem yang dikembangkan berbasis Software as a Service (SaaS), sistem ini mengimplementasi Fungsi SHA-1 untuk tanda tangan digital pada dokumen elektronik [11]. SHA-1 menerima masukan berupa string dengan ukuran maksimal 264 bit. Untuk setiap pesan string, SHA-1 akan menghasilkan keluaran sebanyak 160 bit dari string tersebut dan string keluaran itu disebut message digest [17]. Serangan kebocoran proteksi crpytoanalytic pada fungsi Hash SHA-1 terus meningkat [12], sehingga

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

National Institute of Standard and Technology (NIST) akan bertransisi dari penggunaan SHA-1 sebagai metode proteksi kriptografi untuk semua aplikasi sampai 31 Desember 2030 [18].

b. Advanced Encryption Standard (AES)
Sistem yang dikembangkan oleh [13] melakukan penerapan tanda tangan digital menggunakan Quick Response Code (QR Code) dengan Algoritma Advanced Encryption Standard (AES) pada dokumen permintaan barang.

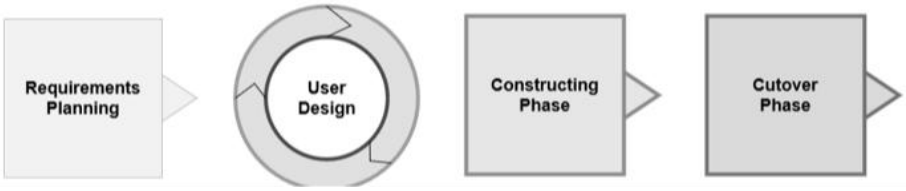
c. Rivest-Shamir-Adleman (RSA)
Sistem yang dikembangkan oleh [14] melakukan penerapan Algoritma RSA pada tanda tangan digital. Sistem penandatanganan digital dimulai dengan melakukan fungsi Hash pada dokumen sehingga menghasilkan message digest. Pada pengembangan sistem ini tidak disebutkan Algoritma Fungsi Hash yang digunakan namun, fungsi Hash yang umum digunakan adalah SHA-256.Pembangkitan kunci publik dan kunci private dilakukan untuk kebutuhan enkripsi dan dekripsi Algoritma RSA. Message digest yang dihasilkan kemudian dienkripsi menggunakan kunci publik dari algoritma RSA yang telah dibangkitkan. Hasil enkripsi ini yang digunakan menjadi tanda tangan digital. Proses verifikasi dilakukan dengan mendekripsi tanda tangan digital menggunakan kunci privat yang sudah dibangkitkan dan membandingkan dengan message digest dari dokumen. Apabila hasilnya bersesuaian, maka dokumen dinyatakan valid, sebaliknya jika hasil dekripsi dengan message digest dokumen tidak bersesuaian maka dokumen tersebut dinyatakan tidak valid. Tanda tangan digital diterapkan pada plaintext yang diketikkan langsung di aplikasi. Sistem yang dikembangkan memiliki arsitektur yang sederhana dengan menggunakan fungsi Hash dan Enkripsi. Sistem memiliki keterbatasan akses untuk memverifikasi keaslian dokumen karena constraint dari maksud di bangkitkannya private key pada Algoritma RSA.

2.2 Analisis Kebutuhan

Tahapan kedua metodologi penelitian ini adalah analisis kebutuhan yang difokuskan pada indikator usability. Setelah merinci dasar teoritis melalui studi literatur, penelitian melangkah ke tahap untuk mendapatkan wawasan lebih spesifik tentang kebutuhan pengguna berdasarkan indikator usability [19]. Tahapan ini melibatkan pengumpulan hasil dari penelitian terdahulu yang relevan untuk memberikan konteks dan pemahaman yang lebih baik. Dalam proses ini, berbagai sumber informasi akan dianalisis untuk mengidentifikasi tren dan perkembangan terbaru. Fokus utama analisis kebutuhan adalah untuk mengetahui teknologi yang tepat yang dapat di implementasikan dalam penelitian ini. Dengan pemahaman yang jelas tentang kebutuhan, diharapkan solusi yang dihasilkan akan lebih efektif dan relevan dengan tantangan yang ada saat ini.

2.3 Perancangan Purwarupa

Software Development Lifecycle yang digunakan pada tahap perancangan purwarupa pada penilitian ini adalah metode Rapid Application Development (RAD). Metode RAD pertama kali diperkenalkan oleh James Martin di tahun 90an. RAD menekankan pada pengembangan perangkat lunak dan input dari pengguna dibanding perencanaan yang teliti dan spesifikasi kebutuhan perangkat lunak [20].



Gambar 2. Model SDLC Rapid Application Development

Adapun beberapa tahapan di dalam metode RAD adalah sebagai berikut :

a. Requirements Planning
Merupakan fase perencanaan dan penentuan perangkat lunak yang akan dikembangkan berdasarkan kebutuhan yang didapat dari hasil analisis terhadap masalah yang diajukan oleh user. Pada tahap ini dilakukan persetujuan terhadap business logic, cakupan, batasan dan persyaratan system yang dibutuhkan selama proses pengembangan perangkat lunak.

b. User Design
Merupakan fase dimana developer melakukan analisis terhadap sistem yang akan dikembangkan dalam bentuk permodelan atau purwarupa sebagai wujud proses sistem dan business logic yang akan dikembangkan. Tahap ini adalah suatu proses interaktif yang terus berlanjut sehingga pengguna bisa mengerti, memodifikasi sehingga pengguna bisa menentukan apakah model presentasi yang dikembangkan ini sudah menemui kebutuhan mereka.

c. Constructing Phase
Fase ini berfokus pada pengembangan perangkat lunak dan fase ini juga bersifat interaksi yang berlanjut antara developer dan pengguna agar pengguna bisa memberikan feedback, selain coding pengembangan perangkat lunak, dilakukan juga integrasi unit dan system testing yang memastikan system berjalan sesuai harapan. Pada

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

tahap ini dilakukan coding sebagai bentuk implementasi algoritma Hash dengan SHA-2 dan Enkripsi dengan AES-256 yang diintegrasikan dalam bentuk aplikasi web dan dilakukan pengujian terhadap fungsionalitas perangkat lunak.

d. Cutover Phase
Fase terakhir pada implementasi RAD merupakan konversi data, testing dan pembuatan konfigurasi enviroment yang siap untuk perangkat lunak berpindah ke fase production serta proses pelatihan agar pengguna bisa memanfaatkan perangkat lunak yang telah dikembangkan. pada tahap ini akan dilakukan deployment pada aplikasi web untuk memastikan sistem perangkat lunak handal dan bisa digunakan oleh pengguna.

2.4 Pengujian Usability

Pengujian perangkat lunak akan menggunakan blackbox testing, dimana pengujian perngkat lunak secara manual akan dilakukan untuk setiap skenario dan fiturnya untuk menguji fungsionalitas perangkat lunak. Pengujian selanjutnya adalah penebaran kuisioner kepada pengguna perangkat lunak di Civitas Akademik, untuk menguji usability perangkat lunak.

Penilitian yang dilakukan menggunakan skala likert sehingga setiap jawaban memiliki skor likert; Skor 1: Sangat Tidak Setuju (STS), Skor 2: Tidak Setuju (TS), Skor 3: Biasa Saja (BS), Skor 4: Setuju (S), Skor 5: Sangat Setuju (SS).
Skala Likert memiliki rumus sebagai berikut :

$$\text{Skala Likert} = \text{Jumlah Responden X Skor Likert} \quad (1)$$

Persentase dari skala Likert dapat dihitung dengan menggunakan persamaan berikut :

$$\text{Presentase Skala Likert} = \frac{\text{skala likert jawaban}}{\text{skala likert tertinggi}} \times 100\% \quad (2)$$

2.5 Pelaporan

Tahap terakhir adalah penyusunan laporan, yang menyajikan keseluruhan penelitian dalam bentuk laporan akhir. Laporan ini mencakup ringkasan penelitian, mulai dari latar belakang dan tujuan, hingga hasil yang diperoleh.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Perancangan

Implementasi perancangan perangkat lunak yang dikembangkan dibagi menjadi dua bagian, backend sebagai bagian yang menjalankan segala proses logika bisnis pada perangkat lunak dan frontend sebagai bagian yang mempresentasikan logika bisnis perangkat lunak kepada pengguna agar melalui tampilan User Interface dapat terjadinya interaksi dalam bentuk input dari pengguna dan output dari perangkat lunak. Perangkat lunak berbasis aplikasi web yang telah selesai dikembangkan di deploy pada virtual machine sebagai server agar aplikasi dapat digunakan oleh banyak pengguna.

3.1.1 Backend

Terdapat dua bagian utama dalam logika bisnis yang digunakan pada perangkat lunak ini, yaitu HTTP Handler yang berfungsi untuk mengatur segala bentuk komunikasi berbasis client – server dalam bentuk Application Programming Interface (API) dan Implementasi Fungsi Hash SHA - 256 serta Enkripsi dengan Advanced Encryption Standard (AES - 256). Backend dibangun dengan menggunakan bahasa pemrograman Go.

3.1.1.1 HTTP Handler

HTTP Handler dibuat dengan Go standard library dan web framework Gin, HTTP Handler dibuat dalam bentuk REST API yang bertanggung jawab untuk membuat koneksi dan akses terhadap database dan memberikan output berupa response kepada user.

Tabel 1. Endpoint pada REST API

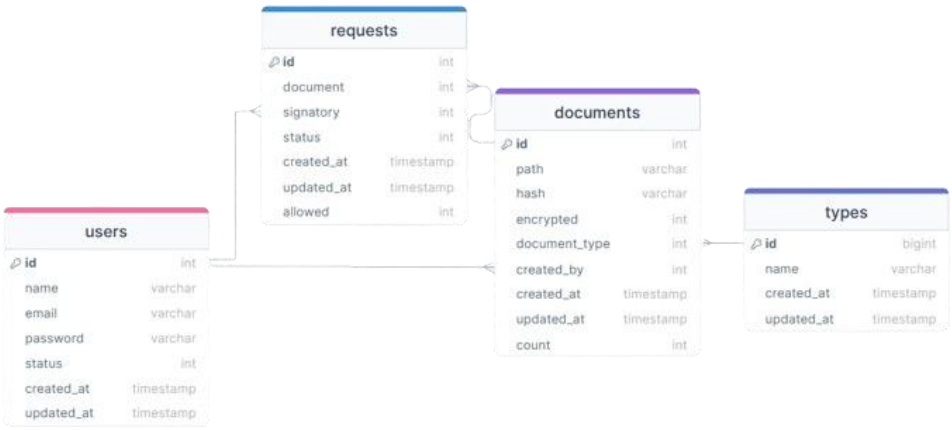
Endpoint	HTTP Method	Content-Type	Route Params
		Public Route	
/login	POST	application / json	-
/download	GET	-	:filename
/decrypt	GET	-	:ciphertext
		Private Route	
/users	GET	-	-
/documents	GET	-	-
/documents	GET	-	:status
/documents	POST	multipart / form-data	-

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

Endpoint	HTTP Method	Content-Type	Route Params
/types	GET	-	-
/types	POST	application / json	-
/requests	GET	-	-
/requests	POST	application / json	-
/requests	PUT	application / json	:requestId
/encryption	PUT	application / json	-

3.1.1.2 ERD Physical Model

ERD physical model untuk menggambarkan hubungan antar data dan informasi yang dipunyai tabel pada suatu database, ERD Physical Model ditunjukan pada Gambar 3.

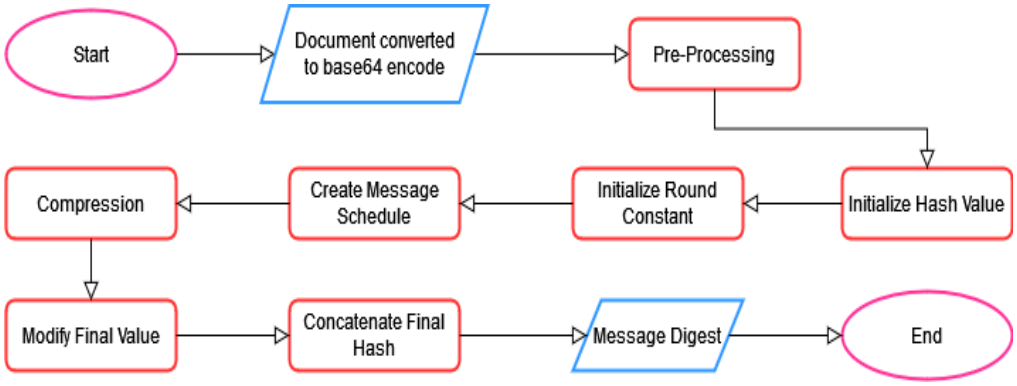


Gambar 3. Entity Relationship Diagram Physical Model

Terdapat empat hubungan antar tabel pada Gambar 3, pada tabel requests attribute signatory memiliki relasi many to one terhadap tabel users dan attribute document memiliki relasi many to one terhadap tabel documents. Pada tabel documents attribute document_type memiliki relasi many to one terhadap tabel types dan attribute created_by memiliki relasi many to one terhadap tabel users.

3.1.1.3 Fungsi Hash & Algoritma Enkripsi

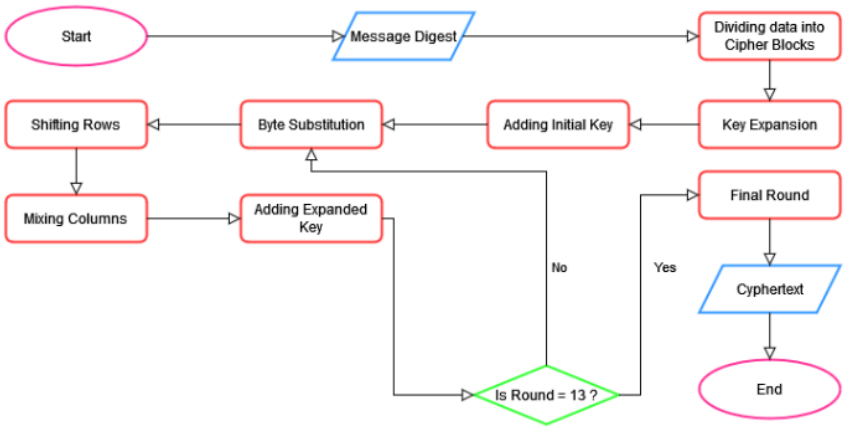
Logika bisnis untuk Fungsi Hash dan Algortima Enkripsi dibuat Go standard library crypto. Fungsi Hash dan Algoritma Enkripsi akan diintegrasikan pada aplikasi web sebagai sebuah service, Gambar 4 menjelaskan Fungsi Hash SHA-256 akan dilakukan ketika dokumen telah selesai di setujui oleh pihak penandatangan, kemudian nilai message digest yang dihasilkan disimpan lalu dilakukan Enkripsi menggunakan AES-256 yang menghasilkan ciphertext untuk dimasukan ke dalam QR Code.



Gambar 4. Proses Hash Dokumen menjadi Message Digest

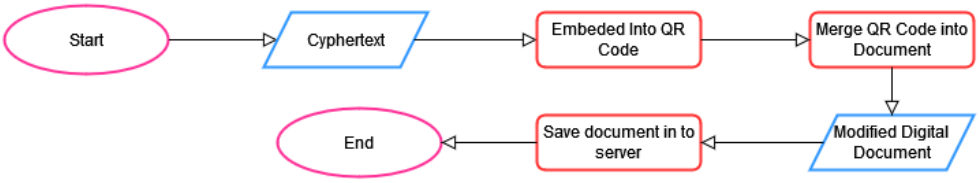
Gambar 5 menjelaskan alur proses enkripsi dimulai dari konversi kunci enkripsi dalam bentuk byte berdasarkan kunci yang telah ditentukan. Selanjutnya dilakukan inisialisasi objek AES berdasarkan nilai kunci yang dibuat dan inisialisasi ukuran byte ciphertext yang akan dihasilkan. Inisialisasi Initialitation Vector (IV) dilakukan sepanjang 16 byte menggunakan random value. Bagian kedua fungsi menjelaskan dilakukannya inisialisasi blok Cipher Block Chaining (CBC) yang merupakan operasi block cipher yang merupakan cara kerja algoritma AES-256 dimana dilakukan operasi enkripsi per 16 block. Fungsi ini menghasilkan ciphertext dalam bentuk byte dan nilai error jika ada.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942



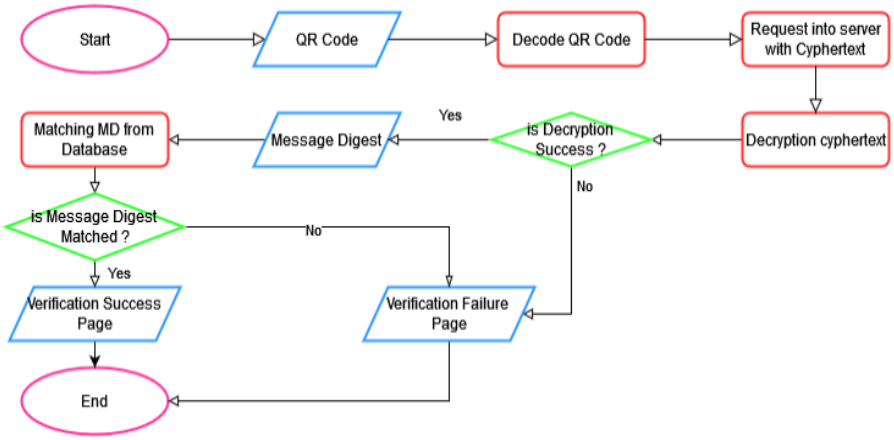
Gambar 5. Proses Enkripsi Message Digest dengan Algoritma AES-256

Gambar 6 menjelaskan proses pembentukan dan penempelan QR-Code pada dokumen, Cipherteks yang dihasilkan akan di encode kedalam QR Code, QR yang dihasilkan akan ditempelkan kedalam dokumen asli pada posisi bawah kanan dan menghasilkan Dokumen digital yang sudah dimodifikasi. Dokumen tersebut akan disimpan pada server dan siap untuk diunduh sesuai kebutuhan.



Gambar 6. Proses Pembentukan dan penempelan QR-Code pada dokumen

Gambar 7 menjelaskan proses verifikasi dokumen digital, langkah pertama yang dilakukan adalah pembacaan nilai kunci yang sama digunakan untuk enkripsi, selanjutnya dilakukan inisialisasi objek AES dengan kunci tersebut. Dilakukan pengecekan panjang ciphertext sebagai input parameter dari fungsi ini untuk memastikan panjang ciphertext harus lebih besar dari ukuran blocksize AES-256. Selanjutnya dilakukan pembagian nilai IV dan ciphertext yang akan didekripsi dimana 16 byte pertama dari ciphertext yang dimasukan nilai IV sedangkan sisanya adalah ciphertext yang akan dilakukan operasi dekripsi menggunakan CBC. Fungsi ini menghasilkan plaintext dalam bentuk byte dan akan mengembalikan nilai error jika ada.



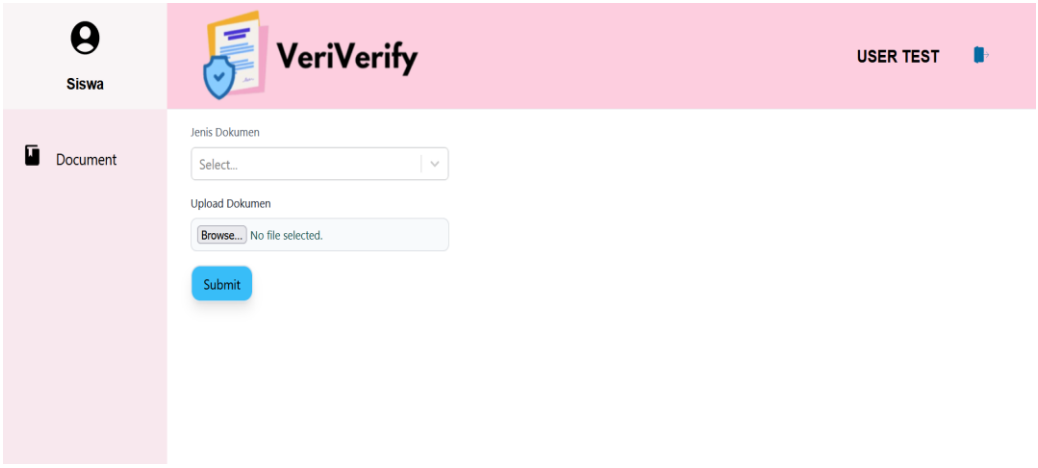
Gambar 7. Proses Verifikasi Dokumen Digital

3.1.2 Frontend

Logika bisnis yang telah dibuat akan direpresntasikan kedalam bentuk interface aplikasi berbasis web, tampilan akan dibuat dengan menggunakan HyperText Markup Language (HTML), Cascade Styling Sheet (CSS) dan Library Javascript. Berikut adalah tampilan antarmuka aplikasi yang telah dibuat :

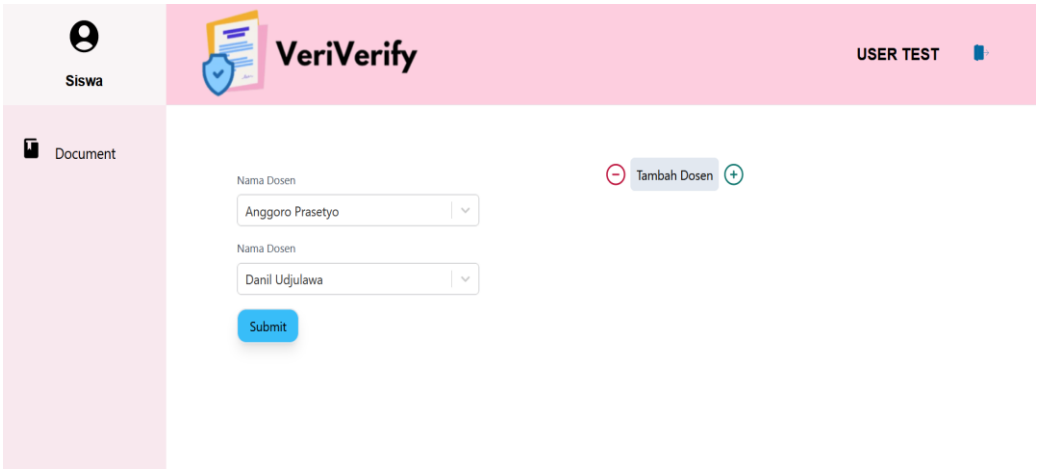
Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

a. Tampilan Tambah Dokumen
Halaman tambah document akan ditampilkan setelah pengguna mengklik button tambah dokumen pada halaman document. Gambar 8 menunjukan tampilan halaman tambah document.



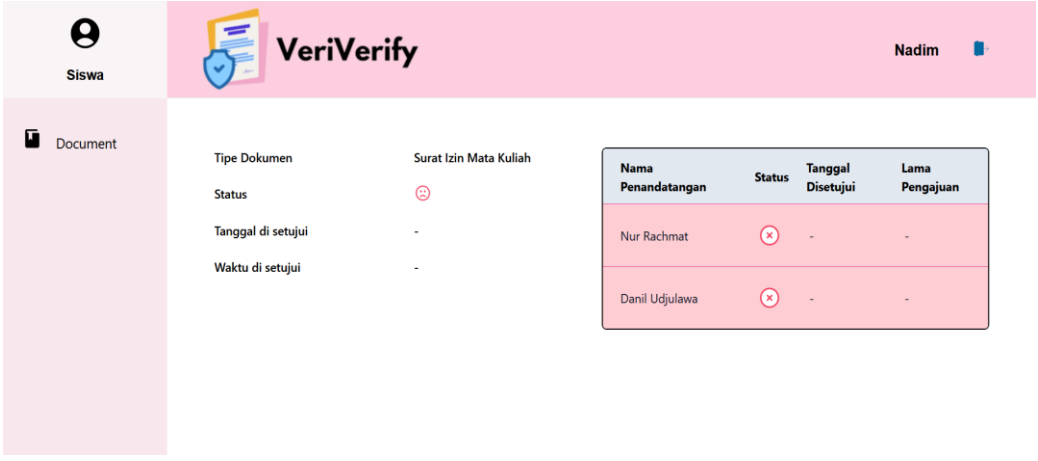
Gambar 8. Tampilan halaman tambah document

Selanjutnya pengguna akan dialihkan ke tampilan yang menunjukan informasi penandatangan yang diinginkan, tampilan ini akan ditunjukan pada Gambar 9, setelah selesai proses submit, pengguna akan dikembalikan pada halaman list document.



Gambar 9. Tampilan halaman pemilihan penandatangan

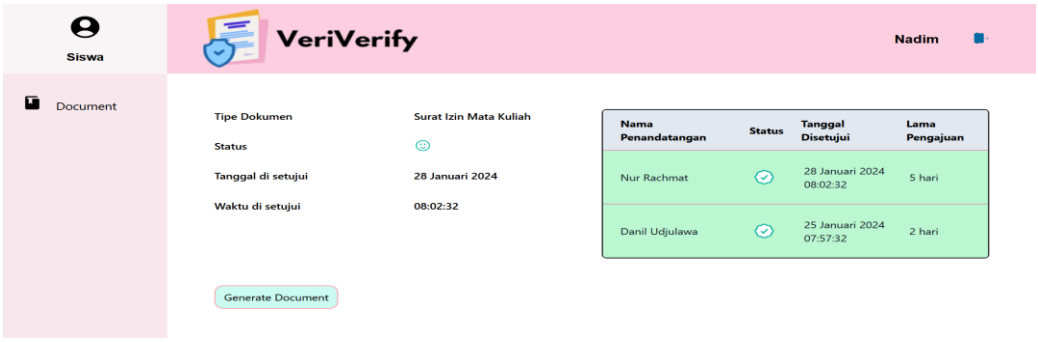
b. Tampilan Detail Dokumen
Halaman detail document akan menampilkan status dokumen saat ini, tanggal dokumen disetujui dan nama - nama penandatangan beserta statusnya. Tampilan halaman detail document yang belum disetujui akan ditunjukan pada Gambar 10.



Gambar 10. Tampilan halaman detail dokumen dengan status belum disetujui

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

Tampilan halaman detail document yang sudah disetujui ditunjukan pada Gambar 11.



Gambar 11. Tampilan halaman detail dokumen dengan status sudah disetujui

Tombol Generate Document pada tampilan halaman detail dokumen yang telah disetujui akan melakukan fungsi hash pada dokumen dan keluarannya akan disimpan di basis data, lalu output fungsi hash berupa message digest akan dienkripsi dan ciphertext yang dihasilkan akan disematkan ke dalam QR Code beserta base URL . Contoh link yang telah dibuat adalah :
http://20.198.254.61/decrypt/4e6d7fe68bc7c47ed7c30f5efc1ee3b9e84d32cdcfdd237922a6e3f499419b3b3066a9d0914b97b488e0243f5b3be423
http://20.198.254.61 adalah base URL dengan endpoint /decrypt dan route parameter yang diisi dengan nilai ciphertext yang merupakan hasil enkripsi nilai hash suatu dokumen yang telah ditanda tangani secara digital. Semua informasi ini akan disematkan kedalam QR Code seperti pada Gambar 12, selanjutnya sistem akan membuatkan lembar pengesahan berisi QR Code tersebut dan akan disisipkan sebagai halaman pertama dengan file dokumen yang telah diunggah oleh pengguna.

c. Tampilan Lembar Pengesahan yang di generate
Tampilan lembar pengesahan pada Gambar 12 berisi QR Code yang diisi dengan alamat base URL dan hasil enkripsi. Halaman lembar pengesahan akan disisipkan pada file PDF yang di upload oleh pengguna ketika melakukan permintaan persetujuan dokumen dan akan menjadi halaman pertama dari file PDF tersebut.

Gambar 12. Tampilan lembar pengesahan

d. Tampilan Hasil Dekripsi
Tampilan hasil Dekripsi muncul setelah QR Code di pindai dan akan diarahkan ke halaman hasil Dekripsi yang menunjukan status verifikasi dokumen tersebut. Tampilan hasil Dekripsi yang sukses akan ditunjukan pada Gambar 13 dan tampilan hasil Dekripsi yang gagal akan ditunjukan pada Gambar 14.

Gambar 13. Tampilan hasil dekripsi sukses

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

Gambar 14. Tampilan hasil dekripsi gagal

3.2 Pengujian

3.2.1 Pengujian Algoritma

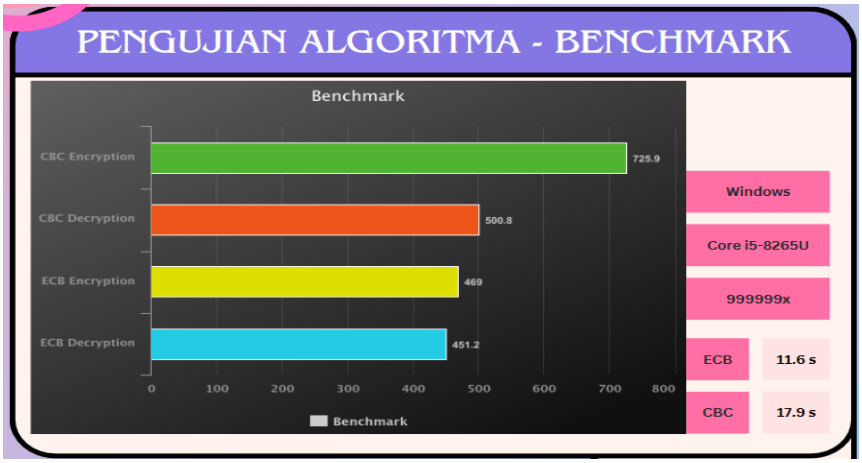
Advanced Encryption Standard (AES) - 256 merupakan standar enkripsi yang menggunakan algoritma Rijndael dengan ukuran blok cipher 128 bit dan ukuran kunci 256 bit. Penilitian ini menggunakan plaintext berupa message digest hasil fungsi hash SHA-256 yang memiliki panjang 32 byte.

Ada berbagai macam mode operasi dan variasi pada Advanced Encryption Standard (AES), secara default AES menggunakan mode operasi Electronic Codebook (ECB). AES dengan mode operasi ECB memiliki kelemahan untuk semua plaintext yang sama maka akan menghasilkan ciphertext yang sama, hal ini berpotensi membocorkan informasi mengenai plaintext karena ada pattern yang bisa di eksploitasi oleh penyerang. Ilustrasi mode operasi ECB akan ditunjukan pada Gambar 11.

Plaintext akan dibagi menjadi ukuran n-block dengan ukuran block 16 byte dan dilakukan Enkripsi dengan AES untuk setiap block tersebut. Setiap block nya bersifat tidak bergantung pada block lainnya sehingga operasi bisa dilakukan secara parallel.

Mode operasi Cipher Block Chaining (CBC) memanfaatkan Initialization Vector (IV) dengan nilai random untuk dilakukan operasi XOR pada plaintext pada block pertama. Mode CBC bersifat bergantung pada block sebelumnya, karena output dari block sebelumnya akan dilakukan operasi XOR dengan block plaintext pada block selanjutnya inilah kenapa mode ini diberi nama Block Chaining. Mode operasi CBC memungkinkan plaintext yang sama persis akan memiliki output yang berbeda untuk setiap operasi sehingga mode ini mampu menutupi kekurangan mode ECB yang meninggalkan pola yang bisa di eksploitasi oleh penyerang. Mode operasi ini memiliki kemungkinan untuk terjadinya block loss atau kegagalan enkripsi pada suatu block yang menyebabkan operasi pada block - block selanjutnya juga gagal.

Projek ini menggunakan AES-CBC Mode dengan alasan bahwa AES-CBC Mode memiliki keamanan yang lebih tinggi dibanding standar default AES-ECB Mode, dilakukan benchmark untuk percobaan enkripsi dan dekripsi sebanyak 9.999.999 kali untuk mendapat waktu pemrosesan operasi. Benchmark dilakukan dengan spesifikasi Operating System Windows dan CPU 8-core Core i5-8265U. Total waktu yang diperlukan untuk AES-ECB Mode adalah 11,6 detik dan AES CBC Mode adalah 17,9 detik. Gambar 15 menunjukan hasil benchmark perbandingan AES-EBC Mode dan AES-CBC Mode.



Gambar 15. Ilustrasi mode operasi AES-CBC

Benchmark diukur dalam satuan ops / second, dimana semakin sedikit ops / second maka dibutuhkan resource yang lebih sedikit untuk menyelesaikan operasi tersebut. Pada Gambar 15 menunjukan bahwa mode AES-CBC membutuhkan ops / second yang lebih banyak dan baik untuk mode AES-ECB dan mode AES-CBC, proses enkripsi membutuhkan resource yang lebih baik dibanding proses dekripsi.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

3.2.2 Pengujian Perangkat Lunak

Pengujian yang dilakukan pada penelitian ini adalah Black Box Testing. Pengujian aplikasi dilakukan dengan cara manual yaitu mengetes semua fitur dan tombol sesuai alur aplikasi.

Tabel 2. Tabel Hasil Pengujian secara Black Box

Aksi yang dilakukan	Hasil yang diharapkan	Hasil Pengujian
Menekan tombol submit pada halaman login dengan posisi form kosong	Pesan error pada label username dan password	Berhasil
Menekan tombol submit pada halaman login dengan posisi form terisi dengan kredensial yang benar	Sukses login dan berpindah ke halaman document	Berhasil
Menekan tombol tambah dokumen	Berpindah ke halaman form pengajuan dokumen	Berhasil
Menekan tombol submit pada halaman pengajuan dokumen dengan posisi jenis dokumen terpilih dan sudah memilih file untuk di upload	Berpindah ke halaman pemilihan pihak penandatangan	Berhasil
Menekan tombol (-) ketika posisi penandatangan satu	Tombol tidak bisa ditekan dan opacity berkurang	Berhasil
Menekan tombol (+) ketika posisi penandatangan empat	Tombol tidak bisa ditekan dan opacity berkurang	Berhasil
Menekan tombol (-) ketika posisi penandatangan lebih dari satu kurang dari empat	Tombol bisa ditekan dan jumlah penandatangan berkurang	Berhasil
Menekan tombol (+) ketika posisi penandatangan lebih dari satu kurang dari empat	Tombol bisa ditekan dan jumlah penandatangan bertambah	Berhasil
Menekan tombol Generate Document pada halaman detail document yang sudah di setujui.	Tombol bisa ditekan dan dialihkan Kembali ke halaman documents	Berhasil
Menekan icon Download pada halaman detail document yang sudah di generate.	Tombol bisa ditekan dan dialog download langsung berjalan lengkap dengan lembar pengesahannya	Berhasil
Menekan tombol Setujui pada halaman detail permintaan pengajuan dokumen	Tombol bisa ditekan dan muncul dialog konfirmasi untuk melakukan persetujuan	Berhasil
Menekan tombol Download pada halaman detail permintaan pengajuan dokumen	Tombil bisa ditekan dan dialog download langsung berjalan sesuai dengan file dokumen yang diajukan	Berhasil

3.2.3 Pengujian Kepuasan Aplikasi

Pengujian kepuasan aplikasi dilakukan dengan menggunakan kuesioner melalui google form. Kuesioner yang dilakukan pada penilitian ini berhubungan dengan tampilan website oleh responden dalam bentuk pertanyaan. Berikut merupakan hasil rekapitulasi perhitungan skala likert kuesioner yang dilakukan ditunjukkan pada Tabel 3.

Tabel 3. Tabel Hasil Pengujian Skala Likert

Pertanyaan terkait Kriteria Penilaian	Persentase Skala Likert	Keterangan
Apakah aplikasi mudah dipahami ?	96 %	Sangat Setuju
Apakah fitur kamera mudah digunakan?	94 %	Sangat Setuju
Apakah fitur ambil dari file mudah digunakan?	90 %	Sangat Setuju
Apakah jenis dan ukuran font mudah dibaca	94 %	Sangat Setuju
Apakah kombinasi warna sudah baik?	94 %	Sangat Setuju
Apakah tata letak tombol sudah baik?	94 %	Sangat Setuju
Apakah dapat mendeteksi dengan baik?	94 %	Sangat Setuju
Rata- rata	93,71 %	Sangat Setuju

Berdasarkan hasil perhitungan persentase yang ditunjukkan pada Tabel 3 maka rata - rata dari persentase diatas adalah 93,71 % maka dapat disimpulkan bahwa aplikasi yang telah dikembangkan dapat berjalan dengan baik.

4. KESIMPULAN

Berdasarkan hasil pengujian secara kuantitatif terhadap algortima yang diimplementasikan pada penelitian ini menggunakan AES-CBC Mode dengan alasan bahwa AES-CBC Mode memiliki keamanan yang lebih tinggi dibanding standar default AES-ECB Mode, dilakukan benchmark untuk percobaan enkripsi dan dekripsi sebanyak 9.999.999 kali untuk mendapat waktu pemrosesan operasi. Benchmark dilakukan dengan spesifikasi Operating

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 229−239
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5942

System Windows dan CPU 8-core Core i5-8265U. Total waktu yang diperlukan untuk AES-ECB Mode adalah 11,6 detik dan AES CBC Mode adalah 17,9 detik.Benchmark diukur dalam satuan ops / second, dimana semakin sedikit ops / second maka dibutuhkan resource yang lebih sedikit untuk menyelesaikan operasi tersebut. Mode AES-CBC membutuhkan ops / second yang lebih banyak dan baik untuk mode AES-ECB dan mode AES-CBC, proses enkripsi membutuhkan resource yang lebih banyak dibanding proses dekripsi. Pengujian dari aspek fungsionalitas yang dilakukan pada penelitian ini adalah Black Box Testing. Pengujian aplikasi dilakukan dengan cara manual yaitu mengetes semua fitur dan tombol sesuai alur aplikasi serta pengujian secara kualitatif dari aspek UI / UX pada purwarupa perangkat lunak maka dapat disimpulkan bahwa sistem perangkat lunak yang telah dikembangkan mampu menjaga integritas dan validitas suatu dokumen yang berbentuk dokumen digital. Perangkat lunak yang dihasilkan memiliki persentase kepuasan sebesar 93,71 %, sehingga dapat disimpulkan bahwa aplikasi yang telah dikembangkan dapat berjalan dengan baik.

REFERENCES

[1] M. Dahlia and W. Susetio, “Tinjauan Yuridis Penggunaan Tanda Tangan Digital Dalam Perjanjian Jual Beli,” J. Multidisiplin Indones., vol. 2, no. 8, pp. 2277–2289, 2023, doi: 10.58344/jmi.v2i8.442.

[2] L. B. Sihombing, “Keabsahan Tanda Tangan Elektronik dalam Akta Notaris,” J. Educ. Dev., vol. 8, no. 1, p. 135, 2020, doi: https://doi.org/10.37081/ed.v8i1.1515.

[3] 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, 2020, doi: 10.29207/resti.v4i6.2502.

[4] V. V. Zubov, “An Electronic Signature Within The Digital Economy,” in Proceedings of the II International Scientific Conference GCPMED 2019 - “Global Challenges and Prospects of the Modern Economic Development,” 2020, pp. 621–625. doi: 10.15405/epsbs.2020.03.89.

[5] Ihwan, “Kasus pemalsuan tanda tangan di Program Pascasarjana UMI tahun 2014 , resmi dilaporkan oleh Lsm Gempa Indonesia.” Accessed: Oct. 05, 2023. [Online]. Available: https://sulselberita.com/2021/10/23/kasus-pemalsuan-tanda-tangan-di-program-pascasarjana-umi-tahun-2014-resmi-dilaporkan-oleh-lsm-gempa-indonesia/

[6] Tempo, “Kasus Pabrik Semen Rembang, Polisi usut kasus pemalsuan tanda tangan.” Accessed: Oct. 05, 2023. [Online]. Available: https://koran.tempo.co/read/berita-utama-jateng/410502/polisi-usut-kasus-pemalsuan-tanda-tangan

[7] Muria, “IEC Palsukan Stempel dan Tanda Tangan Presiden BEM UMK.” Accessed: Oct. 05, 2023. [Online]. Available: https://umk.ac.id/informasi/berita/2100-iec-palsukan-stempel-dan-tanda-tangan-presiden-bem-umk

[8] R. Unila, “Tanggapi Kasus Tanda Tangan Palsu, Rektor Ingatkan Mahasiswa Taat Aturan.” Accessed: Oct. 05, 2023. [Online]. Available: https://www.unila.ac.id/tanggapi-kasus-tanda-tangan-palsu-rektor-ingatkan-mahasiswa-taat-aturan-2/

[9] N. Chalimah and M. Ridhuan, “Menunggu Hasil Kerja Polisi Urusi Pemalsuan Tanda Tangan Gubernur, Penyidik Sedang Menyusun Laporan Terbaru.” Accessed: Oct. 05, 2023. [Online]. Available: https://kaltimpost.jawapos.com/utama/01/02/2023/menunggu-hasil-kerja-polisi-urusi-pemalsuan-tanda-tangan-gubernur-penyidik-sedang-menyusun-laporan-terbaru

[10] F. Z. Abraham, P. I. Santosa, and W. W. Winarno, “Tandatangan Digital Sebagai Solusi Teknologi Informasi Dan Komunikasi (Tik) Hijau: Sebuah Kajian Literatur (Digital Signature As Green Information and Communication Technology (Ict) Solution: a Review Paper),” Masy. Telemat. Dan Inf. J. Penelit. Teknol. Inf. dan Komun., vol. 9, no. 2, p. 111, 2019, doi: 10.17933/mti.v9i2.120.

[11] A. Eritza, M. Ramadhan, and H. Hafizah, “Penerapan Digital Signature Metode SHA dan DSA Pada Slip Gaji Pegawai,” J. Sist. Inf. Triguna Dharma (JURSI TGD), vol. 1, no. 6, p. 906, 2022, doi: 10.53513/jursi.v1i6.6002.

[12] G. Leurent and T. Peyrin, “SHA-1 is a shambles: First chosen-prefix collision on SHA-1 and application to the PGP web of trust,” Proc. 29th USENIX Secur. Symp., pp. 1839–1856, 2020.

[13] W. Pramusinto, B. Trya Sartana, S. Mulyati, and S. Amini, “Implementation of AES-192 Cryptography and QR Code to Verify The Authenticity of Budi Luhur University Student Certificate,” J. Pendidik. Teknol. Kejuru., vol. 3, no. 4, pp. 209–215, 2021, doi: 10.24036/jptk.v3i4.14823.

[14] D. Puspitasari and Y. Permanasari, “Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital,” in Prosiding Matematika, 2020, pp. 14–20. doi: http://dx.doi.org/10.29313/.v0i0.20771.

[15] J. Hutagalung, P. S. Ramadhan, and S. J. Sihombing, “Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature,” J. Teknol. Inf. dan Ilmu Komput., vol. 10, no. 6, pp. 1213–1222, 2023, doi: 10.25126/jtiik.1067319.

[16] L. Ante, “Smart Contracts on the Blockchain – A Bibliometric Analysis and Review,” SSRN Electron. J., no. 10, pp. 1–48, 2020, doi: 10.2139/ssrn.3576393.

[17] R. M. Nasution, “Implementasi Metode Secure Hash Algorithm (SHA-1) Untuk Mendeteksi Orisinalitas File Audio,” Bull. Comput. Sci. Res., vol. 2, no. 3, pp. 73–84, 2022, doi: 10.47065/bulletincsr.v2i3.140.

[18] NIST, “NIST Retires SHA-1 Cryptographic Algorithm.” Accessed: Oct. 05, 2023. [Online]. Available: https://www.nist.gov/news-events/news/2022/12/nist-retires-sha-1-cryptographic-algorithm

[19] H. J. Christanto, “Game Theory Analysis on Marketing Strategy Determination of KAI Access and Traveloka based on Usability of HCI (Human-Computer Interaction),” J. Inf. Syst. Informatics, vol. 4, no. 3, pp. 665–672, 2022, doi: 10.51519/journalisi.v4i3.300.

[20] K. L. D. Mendeja, N. R. Dulce, V. U. Martinez, C. N. Tuazon, N. A. Magnaye, and J. M. Gaspado, “A Development using the Rapid Application Model of peTrace : Peter ’ s Poultry Supply Sales and Monitoring Management System,” Int. J. Metaverse, vol. 1, no. 1, 2023.