JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 13 No. 3S1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v13i3S1.8151

PENGEMBANGAN SISTEM KEAMANAN DATA BERBASIS WEB MENGGUNAKAN KOMBINASI ALGORITMA CHACHA20-POLY1305 DAN ARGON2

Neysa Talitha Jehian[1*], Dedy Kiswanto[2], Muhammad Rizki Andrian Fitra[3], Hansel Valent Evante[4]

[1,2,3,4]Universitas Negeri Medan; Jl. William Iskandar Ps. V, Kenangan Baru Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara 20221; Telp (061) 6614002

Keywords:
Argon2; ChaCha20-Poly1305; Description; Encryption; Zero-Knowledge Encryption.

Corespondent Email:
jxjia05@gmail.com

Abstrak. Penelitian ini bertujuan untuk mengembangkan aplikasi web *Brankas File* yang mampu melakukan proses enkripsi dan dekripsi file secara lokal menggunakan algoritma ChaCha20–Poly1305 dan Argon2/PBKDF2. Sistem dikembangkan dengan pendekatan *client-side encryption*, di mana seluruh proses kriptografi dijalankan di sisi pengguna tanpa keterlibatan server, guna menjaga kerahasiaan data berdasarkan konsep *zero-knowledge encryption*. Metode pengembangan yang digunakan adalah *prototyping*, sedangkan pengujian dilakukan menggunakan *black-box testing* untuk memastikan fungsi sistem berjalan sesuai kebutuhan. Hasil pengujian menunjukkan bahwa seluruh fitur utama, termasuk proses enkripsi, dekripsi, pembangkitan *salt*, serta validasi kekuatan kata sandi, berfungsi dengan baik dengan tingkat keberhasilan 100%. Waktu rata-rata proses dekripsi tercatat sekitar tiga detik, yang menunjukkan efisiensi pemrosesan di sisi klien. Perubahan ukuran file terenkripsi masih berada dalam batas wajar akibat penambahan *metadata*, *nonce*, dan *authentication tag* yang diperlukan untuk menjaga integritas data. Kombinasi ChaCha20–Poly1305 dan Argon2 terbukti efektif dalam menjaga keamanan dan ketahanan terhadap serangan *brute-force*. Selain itu, penerapan fitur *password strength indicator* dan *lockout system* membantu meningkatkan keamanan serta kenyamanan pengguna. Hasil penelitian ini menunjukkan bahwa *Brankas File* memiliki potensi besar sebagai solusi keamanan data berbasis web yang aman, efisien, dan mudah digunakan.

Abstract. *This research aims to develop a File Vault web application capable of performing local file encryption and decryption using the ChaCha20–Poly1305 and Argon2/PBKDF2 algorithms. The system was developed using a client-side encryption approach, where the entire cryptographic process is executed on the user's side without server involvement, in order to maintain data confidentiality based on the concept of zero-knowledge encryption. The development method used is prototyping, while testing is carried out using black-box testing to ensure that the system functions as required. The test results show that all major features, including encryption, decryption, salt generation, and password strength validation, function properly with a 100% success rate. The average decryption time was recorded at around three seconds, which demonstrates the efficiency of client-side processing. Changes in the size of encrypted files are still within reasonable limits due to the addition of metadata, nonce, and authentication tags necessary to maintain data integrity. The combination of ChaCha20–Poly1305 and Argon2 has proven to be effective in maintaining security and resistance to brute-force attacks. In addition, the implementation of the password strength indicator and lockout system features helps improve security and user convenience. The*

results of this study show that Brankas File has great potential as a secure, efficient, and easy-to-use web-based data security solution.

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah mendorong berbagai sektor untuk bertransformasi ke arah digital, termasuk dalam pengelolaan dan pertukaran dokumen elektronik yang kini menjadi standar baru dalam aktivitas administrasi, kontraktual, dan legal [1]. Namun, kemajuan ini juga memunculkan ancaman baru terhadap keamanan data, seperti penyadapan, manipulasi informasi, dan pencurian identitas, yang semakin kompleks seiring perkembangan teknologi [2]. Salah satu metode yang paling efektif untuk melindungi data digital adalah penerapan teknik kriptografi, yaitu ilmu yang mempelajari metode matematis untuk menjaga kerahasiaan, integritas, dan otentikasi informasi [3], [4]. Dalam konteks ini, enkripsi merupakan proses mengubah plaintext menjadi ciphertext, sedangkan dekripsi adalah proses untuk mengembalikan data ke bentuk semula [5], [6].

Meskipun teknik enkripsi telah banyak digunakan dalam menjaga keamanan data, sistem enkripsi berbasis web masih menghadapi tantangan berupa kebutuhan sumber daya yang tinggi saat memproses file besar serta risiko akses data oleh pihak yang tidak berwenang [7]. Selain itu, model keamanan tradisional yang mengandalkan kepercayaan implisit terhadap jaringan internal kini dianggap tidak memadai dalam menghadapi ancaman modern yang semakin dinamis [8].

Beberapa penelitian terdahulu telah menunjukkan efektivitas algoritma kriptografi modern dalam meningkatkan keamanan data digital. Misalnya, penelitian [9] menemukan bahwa algoritma ChaCha20–Poly1305 memiliki efisiensi tinggi dan kompatibilitas dengan TLS 1.3, sementara penelitian lain menyebutkan bahwa Argon2 mampu menurunkan risiko kompromi data hingga 42,5% dibandingkan algoritma hash tradisional [10]. Kedua algoritma tersebut menunjukkan potensi besar dalam pengembangan sistem keamanan data yang kuat dan efisien di lingkungan web.

Berdasarkan hasil-hasil tersebut, penelitian ini bertujuan untuk mengembangkan website enkripsi dan dekripsi file berbasis kombinasi algoritma ChaCha20–Poly1305 dan Argon2, dengan seluruh proses kriptografi dijalankan secara lokal di browser pengguna. Selain menguji efektivitas algoritma dari sisi keamanan, penelitian ini juga menganalisis efisiensi pemrosesan serta kemudahan penggunaan, sehingga diharapkan dapat berkontribusi terhadap pengembangan sistem keamanan web yang praktis, efisien, dan andal di era digital.

2. TINJAUAN PUSTAKA

Ancaman merupakan segala bentuk tindakan yang dapat membahayakan keselamatan, privasi, atau keamanan pihak lain. Dalam konteks digital, hal ini mencakup kejahatan siber seperti penyadapan, pencurian identitas, dan penyalahgunaan data pengguna e-commerce, sehingga diperlukan strategi keamanan yang mampu mengantisipasi serta meminimalkan risiko pelanggaran data [11]. Keamanan data menjadi aspek krusial di era digital untuk melindungi informasi pribadi dan akses terhadap sistem berbasis internet. Salah satu teknik yang umum digunakan adalah hashing, yaitu proses kriptografi yang mengubah data menjadi nilai tetap dengan panjang tertentu untuk menjaga integritas serta keamanan kata sandi [12].

Sebagai respons terhadap meningkatnya kompleksitas ancaman siber, Zero Trust Architecture (ZTA) dikembangkan sebagai paradigma keamanan yang menolak kepercayaan implisit terhadap pengguna, perangkat, dan jaringan internal. Setiap permintaan akses divalidasi berdasarkan identitas serta atribut tertentu sesuai prinsip "never trust, always verify" [13]. Konsep ini sejalan dengan Zero Trust Network Access (ZTNA), yang menggantikan model tradisional berbasis perimeter jaringan yang dianggap aman [14]. ZTA menekankan penerapan autentikasi yang kuat, segmentasi jaringan berbasis identitas, serta pemantauan berkelanjutan untuk mendeteksi ancaman secara real-time [15]. Meskipun demikian,

masih terdapat kesenjangan antara teori dan praktik implementasi model ini, terutama dalam konteks aplikasi web modern [16]. Oleh karena itu, ZTA dinilai sebagai pendekatan keamanan yang relevan di era digital karena mampu memperkuat perlindungan terhadap akses data yang tidak sah dan aktivitas berisiko di lingkungan web maupun jaringan internal.

Enkripsi dan deskripsi data merupakan proses utama dalam menjaga kerahasiaan informasi. Enkripsi adalah proses mengubah pesan asli (plaintext) menjadi bentuk kode atau ciphertext yang tidak dapat dimengerti tanpa kunci tertentu, sedangkan deskripsi merupakan proses kebalikannya, yaitu mengembalikan ciphertext ke bentuk aslinya agar dapat dibaca kembali [17]. Teknologi enkripsi telah banyak dimanfaatkan untuk melindungi komunikasi digital di berbagai sektor, meskipun penerapannya umumnya dilakukan oleh organisasi atau individu dengan kebutuhan tinggi terhadap kerahasiaan data [18].

Dalam bidang kriptografi modern, berbagai algoritma telah dikembangkan untuk meningkatkan keamanan dan efisiensi proses enkripsi. Salah satunya adalah ChaCha20–Poly1305, algoritma Authenticated Encryption with Associated Data (AEAD) yang berfungsi menjaga kerahasiaan, integritas, dan keaslian data secara bersamaan. ChaCha20 merupakan algoritma stream cipher simetris yang mengenkripsi data melalui operasi XOR antara bit plaintext dan keystream, dengan keystream yang dihasilkan dari empat masukan utama: konstanta, kunci, counter, dan nonce menggunakan operasi ARX (Addition, Rotation, XOR) [19]. ChaCha20 dikenal memiliki efisiensi tinggi, terutama pada perangkat bergerak dan aplikasi real-time, meskipun belum banyak diuji pada sistem komunikasi interaktif seperti aplikasi chat [20]. Poly1305, yang menjadi pasangan ChaCha20, berfungsi menghasilkan authentication tag sepanjang 16 byte menggunakan kunci sekali pakai untuk memverifikasi integritas data [21].

Selain itu, Argon2 digunakan sebagai fungsi derivasi kunci sekaligus perlindungan terhadap serangan brute force pada proses autentikasi pengguna. Algoritma ini merupakan metode password hashing yang memanfaatkan sumber daya CPU dan memori secara intensif untuk memperlambat pemrosesan paralel oleh penyerang. Argon2 memiliki tiga varian yaitu Argon2i, Argon2d, dan Argon2id yang masing-masing dioptimalkan untuk menghadapi serangan waktu dan side channel. Algoritma ini juga dinobatkan sebagai pemenang Password Hashing Competition (PHC) karena efisiensi serta ketahanannya dalam sistem keamanan modern [22].

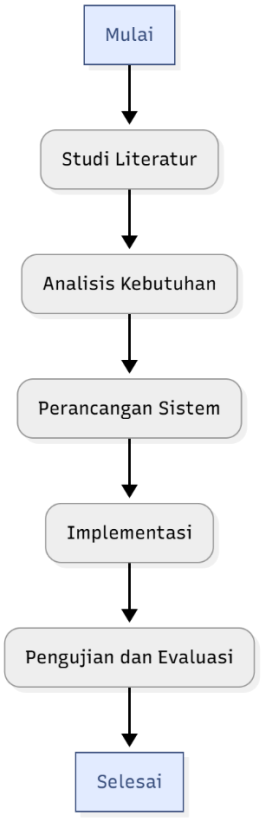
Kombinasi algoritma ChaCha20–Poly1305 dan Argon2 dianggap mampu memberikan tingkat keamanan yang tinggi dalam melindungi data digital. ChaCha20–Poly1305 berperan menjaga kerahasiaan dan integritas data, sedangkan Argon2 memperkuat sistem terhadap serangan berbasis kata sandi. Kolaborasi keduanya memungkinkan penerapan sistem keamanan yang efisien, fleksibel, dan kompatibel lintas platform, khususnya pada aplikasi berbasis web.

3. METODE PENELITIAN

3.1 Metode Pengembangan Sistem

Penelitian ini menggunakan pendekatan metode pengembangan sistem berbasis prototyping, yang menekankan pada iterasi berulang antara perancangan dan evaluasi hingga sistem mencapai fungsionalitas yang diharapkan. Metode ini dipilih karena memungkinkan pengembang untuk memperoleh umpan balik langsung dari pengguna dan melakukan penyempurnaan terhadap desain antarmuka maupun fungsi enkripsi secara cepat dan efisien.

Tahap implementasi sistem dilakukan melalui beberapa langkah terstruktur yang mencakup analisis kebutuhan, perancangan, implementasi, serta pengujian dan evaluasi. Setiap tahap saling berkaitan untuk memastikan bahwa sistem kriptografi yang dibangun dapat berfungsi secara optimal, aman, dan mudah digunakan.



Gambar 1. Tahapan penelitian

Pada tahap analisis kebutuhan, dilakukan identifikasi terhadap kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional meliputi kemampuan untuk melakukan proses enkripsi dan dekripsi file secara lokal di browser, sedangkan kebutuhan non-fungsional mencakup keamanan password melalui derivasi kunci menggunakan Argon2 atau PBKDF2, serta kemudahan penggunaan dengan antarmuka web yang responsif dan intuitif.

Tahap berikutnya adalah perancangan sistem (design). Pada tahap ini dirancang arsitektur sistem, desain antarmuka pengguna (UI), serta alur proses enkripsi dan dekripsi menggunakan algoritma ChaCha20-Poly1305. Perancangan dilakukan dengan memperhatikan efisiensi proses, keamanan data, dan pengalaman pengguna yang sederhana namun fungsional.

Tahap implementasi dilakukan menggunakan teknologi HTML5, Tailwind CSS, dan JavaScript (ES6+) di sisi klien (client-side). Seluruh proses kriptografi dijalankan di browser menggunakan Web Crypto API, dengan fungsi-fungsi utama seperti importKey(), deriveBits(), dan getRandomValues() untuk menghasilkan kunci acak, derivasi kunci, serta proses enkripsi dan dekripsi file.

Tahap terakhir adalah pengujian dan evaluasi, yang bertujuan untuk memastikan seluruh komponen sistem berfungsi dengan benar. Pengujian mencakup validasi hasil enkripsi dan dekripsi, serta verifikasi terhadap mekanisme keamanan seperti password hashing, lockout system, dan password strength meter. Hasil evaluasi digunakan untuk menilai tingkat keamanan dan keandalan sistem sebelum diimplementasikan secara penuh.

3.2 Arsitektur Sistem

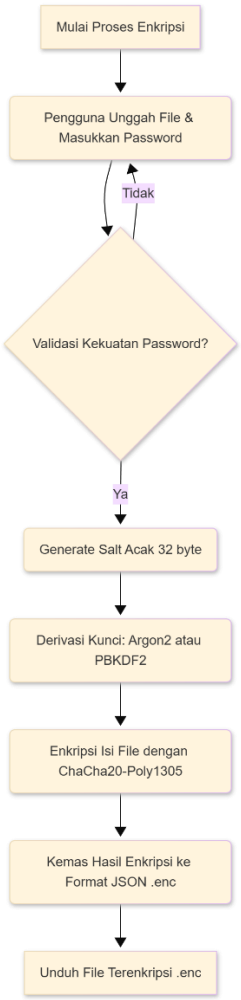
Arsitektur sistem dirancang dengan pendekatan client-side encryption, di mana seluruh proses kriptografi dilakukan langsung di sisi pengguna tanpa mengirimkan data ke server. Pendekatan ini menerapkan prinsip zero-knowledge encryption, sehingga password dan kunci hasil derivasi tidak pernah tersimpan atau dikirim dalam bentuk plaintext. Secara umum, sistem terdiri atas tiga komponen utama, yaitu antarmuka pengguna yang menyediakan fitur unggah file, input password, dan tombol aksi enkripsi maupun dekripsi; fungsi inti yang menangani pembacaan file, derivasi kunci, validasi kekuatan password, serta proses enkripsi-dekripsi; dan lapisan keamanan yang mencakup hashing password dengan PBKDF2, pembangkitan salt acak sepanjang 32 byte, serta sistem penguncian otomatis (lockout system) setelah tiga kali percobaan gagal. Dengan struktur ini, seluruh proses berjalan terpisah namun tetap terintegrasi untuk menjaga keamanan dan kerahasiaan data pengguna.

3.3 Algoritma Kriptografi

Aplikasi ini memanfaatkan kombinasi dua algoritma utama, yaitu ChaCha20-Poly1305 dan Argon2/PBKDF2. ChaCha20-Poly1305 digunakan sebagai authenticated encryption yang tidak hanya menjaga kerahasiaan data tetapi juga memastikan integritasnya melalui keystream enkripsi (ChaCha20) dan verifikasi message authentication code (Poly1305). Algoritma ini juga dikenal lebih tahan terhadap berbagai jenis serangan kriptanalisis serta tidak bergantung pada akselerasi perangkat keras, sehingga tetap fleksibel di berbagai platform [23]. Sementara itu, Argon2 digunakan untuk

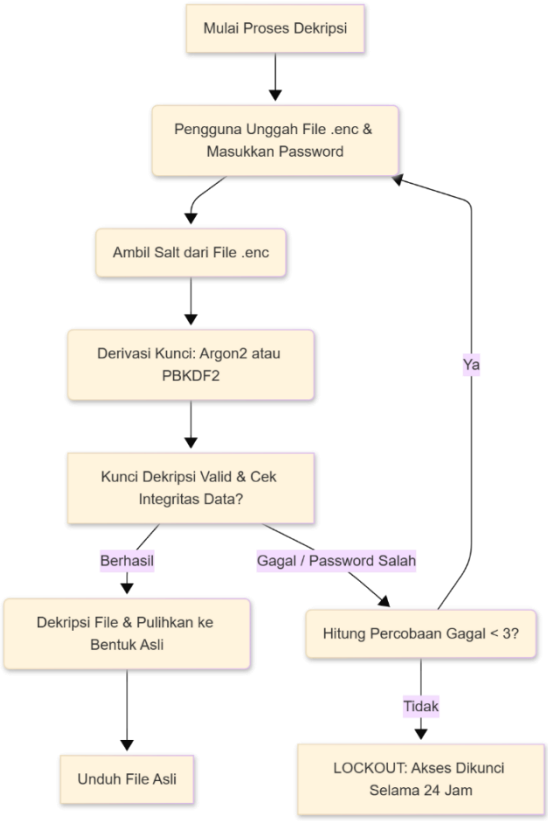
derivasi kunci dari password karena kemampuannya mengoptimalkan penggunaan CPU dan memori secara bersamaan, yang secara signifikan meningkatkan biaya bagi penyerang dalam melakukan serangan brute-force [22]. Kombinasi ini menghasilkan sistem yang aman, efisien, dan kompatibel lintas platform.

3.4 Prosedur Implementasi



Gambar 2. Flowchart proses enkripsi file

Proses implementasi sistem dilakukan secara bertahap, dimulai dari pengunggahan file hingga proses enkripsi atau dekripsi selesai. Saat pengguna mengunggah file, sistem melakukan hashing terhadap password menggunakan Argon2 atau PBKDF2 yang dikombinasikan dengan salt acak sepanjang 32 byte. Hasil derivasi kunci tersebut digunakan oleh algoritma ChaCha20-Poly1305 untuk mengenkripsi isi file. Output enkripsi dikemas dalam format JSON yang memuat data terenkripsi, salt, serta metadata tambahan, lalu dikonversi menjadi file berekstensi .enc agar dapat diunduh pengguna.



Gambar 3. Flowchart proses dekripsi file

Gambar 3 menunjukkan alur proses dekripsi file yang dilakukan oleh sistem. Proses dimulai ketika pengguna mengunggah file berekstensi .enc dan memasukkan password, kemudian sistem mengambil salt, melakukan derivasi kunci menggunakan Argon2 atau PBKDF2, serta memverifikasi integritas data untuk memastikan keaslian file. Jika kunci dekripsi valid maka sistem akan memulihkan data ke bentuk aslinya, sedangkan apabila terjadi kesalahan password lebih dari tiga kali maka akses akan terkunci selama 24 jam. Seluruh proses kriptografi berlangsung secara lokal di browser tanpa melibatkan server sehingga kerahasiaan data sepenuhnya terjaga di sisi pengguna.

3.5 Pengujian Sistem

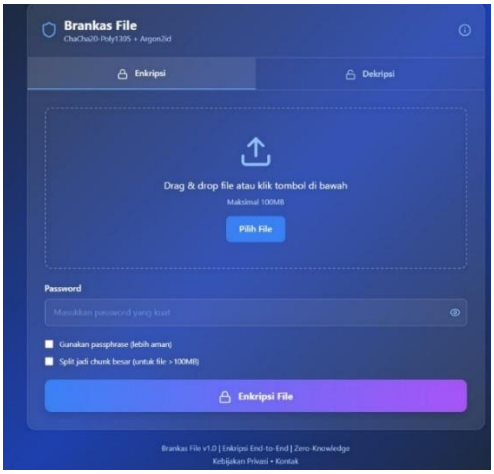
Tahap pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi dan memiliki tingkat

keamanan yang memadai. Metode pengujian yang digunakan adalah black-box testing, di mana pengujian difokuskan pada keluaran sistem berdasarkan masukan tertentu tanpa melihat struktur kode internal. Pengujian dilakukan terhadap beberapa aspek, antara lain validasi proses enkripsi dan dekripsi, pengujian kekuatan mekanisme password hashing dan sistem lockout, serta pengujian antarmuka untuk memastikan tampilan dan fungsionalitas berjalan dengan baik pada berbagai perangkat. Hasil pengujian digunakan sebagai dasar evaluasi dan penyempurnaan sistem, sehingga aplikasi yang dikembangkan dapat memberikan performa optimal sekaligus menjaga keamanan data pengguna secara menyeluruh.

4. HASIL DAN PEMBAHASAN

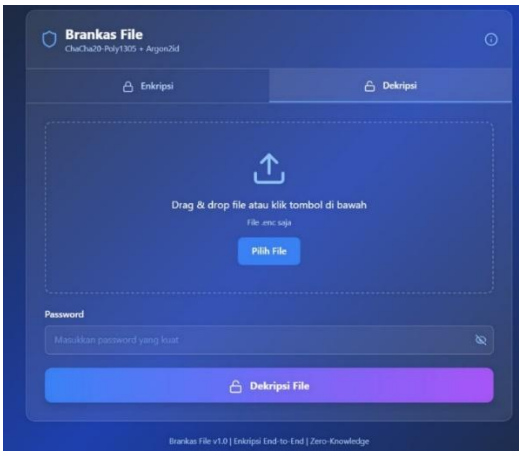
Hasil implementasi penelitian ini berupa sebuah aplikasi web bernama Brankas File yang berfungsi untuk melakukan proses enkripsi dan dekripsi file secara lokal menggunakan algoritma ChaCha20-Poly1305 dan Argon2/PBKDF2. Aplikasi dikembangkan berbasis client-side encryption, di mana seluruh proses kriptografi dilakukan langsung pada browser pengguna tanpa keterlibatan server. Sistem ini mengadopsi konsep zero-knowledge encryption, sejalan dengan prinsip Zero-Knowledge Protocol (ZKP) yang memungkinkan verifikasi data tanpa mengungkapkan informasi yang mendasarinya [24], sehingga password dan kunci hasil derivasi tidak pernah dikirimkan ataupun disimpan di sisi server.

Antarmuka sistem dibangun menggunakan HTML5, Tailwind CSS, dan JavaScript (ES6+). Pengguna dapat memilih mode enkripsi atau dekripsi melalui tab switcher, kemudian mengunggah file melalui area drag and drop atau file picker, serta memasukkan password sebagai kunci utama. Proses enkripsi menghasilkan file baru dengan ekstensi .enc, sedangkan proses dekripsi mengembalikan file ke bentuk aslinya apabila password yang dimasukkan sesuai dengan yang digunakan saat enkripsi.



Gambar 4. Tampilan menu enkripsi pada aplikasi brankas file

Gambar 4 menunjukkan tampilan antarmuka utama pada mode Enkripsi. Pengguna dapat mengunggah file melalui area drag & drop atau dengan tombol Pilih File. Sistem juga menyediakan opsi tambahan seperti penggunaan passphrase untuk keamanan lebih tinggi dan split chunk untuk memproses file berukuran besar.



Gambar 5. Tampilan Menu Dekripsi pada Aplikasi Brankas File

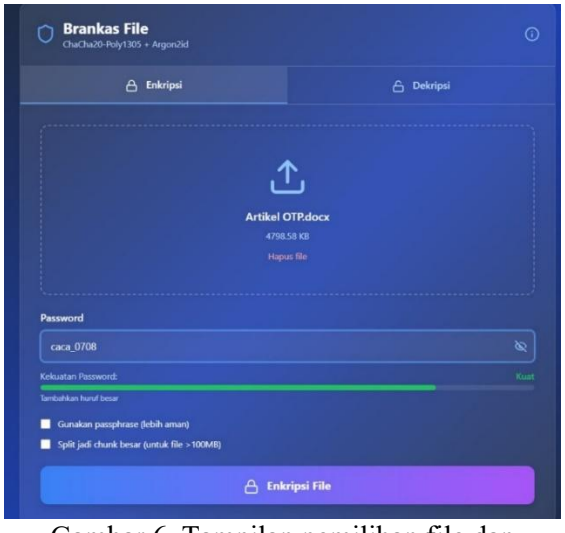
Gambar 5 menampilkan halaman Dekripsi yang digunakan untuk memulihkan file terenkripsi ke bentuk aslinya. Pengguna hanya perlu memilih file dengan ekstensi .enc dan memasukkan password yang sama dengan saat proses enkripsi.

Pengujian dilakukan menggunakan metode black-box testing untuk memastikan fungsi-fungsi utama sistem berjalan sesuai dengan kebutuhan. Metode ini berfokus pada pengujian perilaku fungsional perangkat lunak tanpa

memperhatikan struktur internal program, dengan tujuan memastikan setiap fitur bekerja sesuai spesifikasinya [25]. Berdasarkan hasil uji coba, seluruh fitur inti seperti proses enkripsi, dekripsi, pembangkitan salt, serta validasi kekuatan password telah berfungsi dengan baik. Sistem berhasil menghasilkan keluaran file terenkripsi dan mampu melakukan dekripsi secara sempurna menggunakan password yang sama.

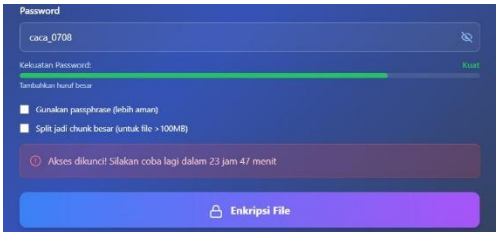
Hasil pengujian menunjukkan bahwa proses kriptografi inti berjalan dengan baik dan transparan bagi pengguna. Pada tahap awal, sistem menampilkan indikator kekuatan password secara real-time sebelum proses enkripsi dimulai. Fitur ini memastikan pengguna dapat memilih kata sandi yang memenuhi standar keamanan minimum untuk melindungi file yang akan diproses. Tampilan indikator tersebut membantu pengguna memahami tingkat keamanan dari kombinasi karakter yang digunakan, termasuk panjang password, variasi karakter, dan penggunaan simbol.

Untuk menguji fungsionalitas utama sistem, dilakukan serangkaian uji coba terhadap proses enkripsi dan dekripsi file. Pengujian ini bertujuan untuk memastikan bahwa algoritma kriptografi yang diterapkan, yaitu ChaCha20-Poly1305 dan PBKDF2-SHA256, mampu berjalan dengan baik dalam menghasilkan file terenkripsi yang valid serta dapat dikembalikan ke bentuk aslinya.



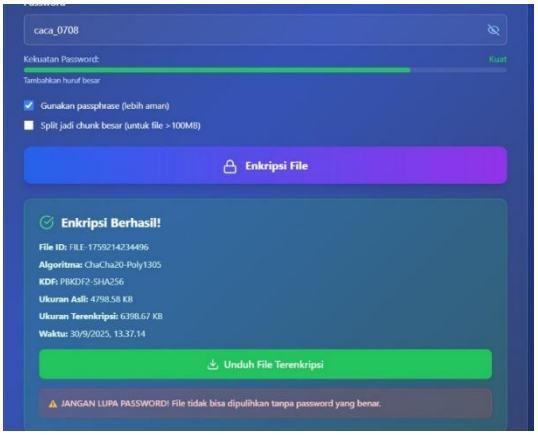
Gambar 6. Tampilan pemilihan file dan validasi sandi

Gambar 6 menunjukkan proses awal sebelum enkripsi dilakukan. Sistem menampilkan indikator kekuatan password secara real-time, sehingga pengguna dapat memastikan kata sandi memenuhi standar keamanan.



Gambar 7. Tampilan fitur lockout

Pada gambar 7 menunjukkan uji coba terhadap fitur lockout system. Setelah tiga kali percobaan password salah, sistem secara otomatis menonaktifkan tombol enkripsi selama 24 jam untuk mencegah serangan brute force.



Gambar 8. Hasil proses enkripsi berhasil

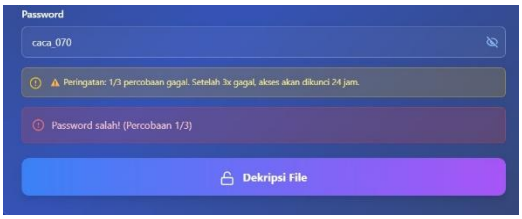
Gambar 8 menunjukkan hasil enkripsi berhasil dilakukan dengan algoritma ChaCha20-Poly1305 dan derivasi kunci PBKDF2-SHA256. Informasi seperti ukuran file asli, ukuran terenkripsi, serta waktu pemrosesan ditampilkan secara transparan di antarmuka.

77a764dc3fc584cebb8787a23b5","passwordHash":"a137d8996948e9175d3594d19010e0588415ace7a8a4faff93e917f53f6369f1","wrappedKey":"ODQzMjM1NGIxOTAyZGFjNzIyNDkxNZFhODBlYzQ4YTE0OGUxYzQxN2ViYjE2YmMw","originalName":"Artikel OTP.docx","originalSize":4913741,"originalType":"application/vnd.openxmlformats-officedocument.wordprocessingml.document","timestamp":"2025-09-30T06:37:14.389Z","usePassphrase":true,"useLargeChunk":false,"encryptedContent":"UEsDBBQABgAIAAAAIQAuCc0EGAIAAG0NAAATAAgCW0NvbnRlbnRfVHlwZXNdLnhtbCCiBAIooAACAAAA

Gambar 9. Struktur data output mentah konten file terenkripsi (.enc)

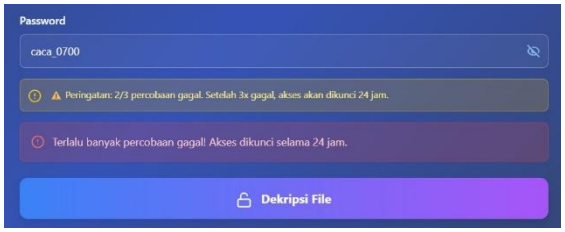
Setelah proses enkripsi berhasil, file asli (Artikel OTP.docx) diubah menjadi file keluaran dengan ekstensi .enc. Konten dari file terenkripsi ini bukan lagi file dokumen biner biasa, melainkan struktur data JSON terenkripsi yang menyimpan metadata kriptografi dan konten file yang dienkripsi (dalam format Base64).

Setelah proses enkripsi dinyatakan berhasil, langkah selanjutnya adalah melakukan pengujian terhadap proses dekripsi untuk memastikan bahwa sistem mampu memulihkan file terenkripsi menggunakan password yang sesuai. Pengujian ini juga digunakan untuk memverifikasi efektivitas algoritma dalam menjaga integritas data serta menguji mekanisme keamanan tambahan seperti fitur lockout system.



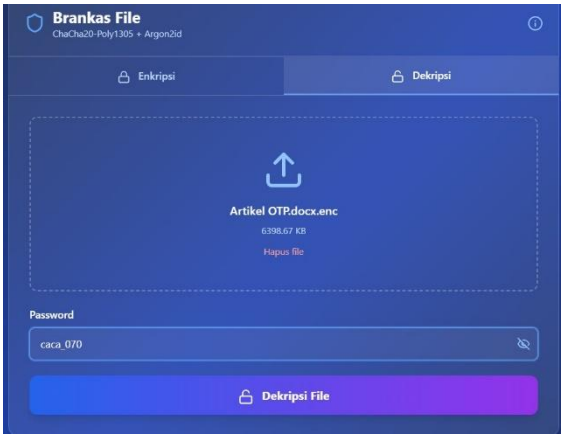
Gambar 10. Percobaan pertama password salah

Gambar 10 menunjukkan pesan peringatan "Password salah! (Percobaan 1/3)" dan sistem mencatat satu kegagalan, memberikan peringatan bahwa setelah 3 kali gagal, akses akan dikunci selama 24 jam.



Gambar 11. Tampilan fitur lockout (gagal 3/3)

Gambar 11 menunjukkan hasil percobaan ketiga yang gagal, di mana sistem secara otomatis menampilkan pesan "Terlalu banyak percobaan gagal! Akses dikunci selama 24 jam" dan menonaktifkan tombol Dekripsi untuk jangka waktu yang ditentukan.



Gambar 12. Tampilan menu dekripsi

Gambar 12 menunjukkan hasil akhir setelah memasukkan password yang benar, dengan konfirmasi "Dekripsi Berhasil!" dan rincian File Asli (Artikel OTP.docx) serta tombol untuk mengunduh file asli.

Berdasarkan pengujian fungsional menggunakan metode black-box testing, seluruh komponen utama sistem telah berjalan sesuai dengan rancangan. Proses enkripsi dan dekripsi berhasil dilakukan dengan tingkat keberhasilan 100% pada berbagai format file seperti DOCX, PDF, dan TXT, di mana file terenkripsi dapat dikembalikan ke bentuk aslinya secara utuh menggunakan password yang sama. Rata-rata waktu proses dekripsi tercatat sekitar 3 detik, menunjukkan bahwa sistem mampu melakukan pemrosesan data secara efisien di sisi klien. Perubahan ukuran file setelah enkripsi masih berada pada kisaran wajar untuk algoritma AEAD, karena adanya penambahan metadata, nonce, dan authentication tag yang diperlukan untuk menjaga integritas serta otentikasi data. Selain itu, fitur password strength indicator dan lockout system berfungsi optimal dalam membatasi percobaan login berulang serta membantu pengguna memilih kata sandi yang kuat, sehingga sistem dinyatakan stabil dan siap untuk tahap pengujian performa lanjutan pada penelitian berikutnya.

Berdasarkan hasil pengujian, sistem Brankas File terbukti mampu menjalankan proses enkripsi dan dekripsi secara lokal dengan waktu rata-rata sekitar tiga detik. Fitur keamanan tambahan seperti password strength indicator dan lockout system berfungsi dengan baik dalam mencegah percobaan brute-force. Pendekatan client-side encryption yang

diterapkan sesuai dengan prinsip Zero Trust Architecture karena seluruh kunci dan sandi diproses di sisi pengguna tanpa dikirim ke server. Dengan demikian, sistem ini menunjukkan keseimbangan antara keamanan, efisiensi, dan kemudahan penggunaan.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi web Brankas File yang menerapkan algoritma ChaCha20–Poly1305 dan Argon2 untuk proses enkripsi dan dekripsi file secara lokal di browser (client-side encryption). Sistem terbukti aman, efisien, dan mudah digunakan, dengan waktu dekripsi rata-rata sekitar tiga detik dan tingkat keberhasilan 100%. Seluruh proses kriptografi dilakukan di sisi klien tanpa melibatkan server, sehingga mendukung prinsip zero-knowledge dan menjaga kerahasiaan data sepenuhnya di perangkat pengguna.

Penerapan algoritma AEAD ChaCha20–Poly1305 terbukti mampu menjaga integritas serta kerahasiaan data, sedangkan Argon2 memperkuat sistem terhadap serangan brute-force melalui mekanisme derivasi kunci yang adaptif terhadap konsumsi memori dan CPU. Implementasi fitur password strength indicator dan lockout system juga mendukung keamanan pengguna dengan mencegah percobaan login berulang.

Untuk pengembangan selanjutnya, sistem ini dapat ditingkatkan melalui integrasi dengan Zero Trust Architecture (ZTA) dan penerapan Just-In-Time (JIT) Authentication Token guna memperkuat lapisan autentikasi. Selain itu, optimalisasi performa untuk file berukuran besar dan penerapan penyimpanan terenkripsi berbasis cloud juga dapat menjadi fokus penelitian berikutnya agar sistem lebih adaptif terhadap kebutuhan keamanan data modern.

DAFTAR PUSTAKA

[1] M. Patria and D. A. Andriati, "Analisis Komparatif Performa AES-GCM dan ChaCha20-Poly1305 dalam Enkripsi Dokumen PDF Berbasis AEAD," *Arcitech: Journal of Computer Science and Artificial Intelligence*, vol. 5, no. 1, pp. 49–69, 2025.

[2] N. A. Tarigan, A. C. N. Anggreni, A. Balqis, I. Nurfadilah, E. S. Bakti, and F. Mahyudin, "Pengembangan Aplikasi Kriptografi RSA dan SHA-256 Berbasis Web Menggunakan Flask," *JIKUM: Jurnal Ilmu Komputer*, vol. 1, no. 2, pp. 40–44, 2025.

[3] A. Aprizald, M. A. Hasan, and D. Setiawan, "Aplikasi Keamanan Data Berbasis Web Menggunakan Algoritma AES 128 Untuk Enkripsi Dan Dekripsi Data," *JEKIN-Jurnal Teknik Informatika*, vol. 2, no. 2, pp. 85–95, 2022.

[4] A. Ariska and W. Wahyuddin, "Penerapan Kriptografi Menggunakan Algoritma DES (Data Encryption Standard)," *Jurnal Sintaks Logika*, vol. 2, no. 2, pp. 9–19, 2022.

[5] D. Ramalinda and A. R. Raharja, "Strategi Perlindungan Data Menggunakan Sistem Kriptografi Dalam Keamanan Informasi," *Journal of International Multidisciplinary Research*, 2024, doi: 10.62504/jimr679.

[6] J. Lantu, K. Santa, F. Sangkop, and O. Kembuan, "Pengembangan Sistem Enkripsi File Berbasis Web Dengan Menggunakan Metode Advanced Encryption Standard," *Journal of Informatics, Business, Education and Innovation Technology*, vol. 3, no. 5, pp. 97–109, 2025.

[7] M. Irvai and N. Efranda, "Optimalisasi Enkripsi File Menggunakan Algoritma AES-256 Berbasis Web Dengan Integrasi Kompresi Adaptif," *BETRIK*, vol. 15, no. 3, pp. 528–536, 2024.

[8] Z. Cui and Z. Song, "Enterprise Security Incident Analysis and Countermeasures Based on the T-Mobile Data Breach," 2025.

[9] R. Serrano, C. Duran, M. Sarmiento, C. K. Pham, and T. T. Hoang, "ChaCha20–Poly1305 authenticated encryption with additional data for transport layer security 1.3," *Cryptography*, vol. 6, no. 2, p. 30, 2022.

[10] P. Tippe and M. P. Berner, "Evaluating Argon2 Adoption and Effectiveness in Real-World Software," in *International Conference on Availability, Reliability and Security*, Cham: Springer Nature Switzerland, 2025, pp. 25–46.

[11] A. P. Kehista *et al.*, "Analisis Keamanan Data Pribadi pada Pengguna E-Commerce: Ancaman, Risiko, Strategi Kemanan (Literature Review)," *Jurnal Ilmu Manajemen Terapan (JIMT)*, vol. 4, no. 5, 2023.

[12] Y. R. Hutasoit, V. Simangunsong, and D. Siallagan, "Analisis Terhadap Keamanan Password Menggunakan Hash SHA-256," *JURNAL QUANCOM: QUANTUM COMPUTER JURNAL*, vol. 3, no. 1, pp. 13–17, 2025.

[13] I. Zuhrianto and S. R. Astari, "Penerapan Zero Trust Architecture untuk Mitigasi

Ancaman Pembajakan Akun WhatsApp," *JITU : Journal Informatic Technology And Communication*, vol. 9, no. 1, pp. 50–58, 2025, doi: 10.36596/jitu.v9i1.1815.

[14] H. Najwa, "Analisis penerapan Trust Network Access (ZTNA) dengan penggunaan CAPTCHA pada website umum," *Technology Sciences Insights Journal*, vol. 1, no. 2, pp. 76–80, 2024.

[15] I. Muakhori and N. Syamsiah, "Pengamanan Arsitektur Microservices pada Aplikasi Perusahaan: Strategi dan Implementasi," *Info Kripto*, vol. 19, no. 1, pp. 29–37, 2025.

[16] R. W. Darmawan, I. Irawan, and S. Petriansyah, "Analisis Adaptif Zero Trust Architecture (ZTA) Berbasis Machine Learning untuk Deteksi Intrusi pada Jaringan IoT dalam Infrastruktur Kritis," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 3, no. 4, pp. 36–45, 2025.

[17] J. H. Sinaga, M. Pangaribuan, F. Fazly, I. Rivaldo, and I. Gunawan, "Penerapan Enkripsi Dan Deskripsi Menggunakan Algoritma Data Encryption Standart Dengan Pemograman Matlab," *Jurnal Media Informatika*, vol. 4, no. 1, pp. 63–69, 2022, [Online]. Available: https://doi.org/10.55338/jumin.v4i1.468

[18] S. N. Nugraha, "Penerapan Algoritma Kriptografi ElGamal pada Aplikasi Pengamanan Pesan Berbasis Website," *Jurnal Informatika Dan Teknik Elektro Terapan*, vol. 12, no. 3, 2024, [Online]. Available: https://doi.org/10.23960/jitet.v12i3.4794

[19] F. Rahim and Y. R. Nasution, "Implementasi Algoritma ChaCha20 Pada Pengamanan File Citra Bitmap," *JURNAL FASILKOM*, vol. 14, no. 3, pp. 615–626, 2024.

[20] D. Darmansyah and A. H. Hasugian, "Enkripsi Pesan Chat Menggunakan Algoritma Chacha20 Pada Aplikasi Komunikasi Real-Time," *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 10, no. 2, pp. 544–554, 2025.

[21] N. V. M. Thanh, "Implementation of ChaCha20-Poly1305 on Self-Organization Data Framing for Enhancing IoT Communication," *REV Journal on Electronics and Communications*, vol. 14, no. 4, 2024.

[22] S. Eum, H. Kim, M. Song, and H. Seo, "Optimized implementation of ARGON2 utilizing the graphics processing unit," *Applied Sciences*, vol. 13, no. 16, p. 9295, 2023.

[23] A. Susanti, B. A. Prasetiya, O. D. Pangesti, L. D. Suryawati, and I. A. Saputro, "Perbandingan Kinerja dan Keamanan Algoritma Kriptografi Modern AES-GCM dengan ChaCha20-Poly1305," *Infomatek*, vol. 26, no. 2, pp. 253–264, 2024, doi: 10.23969/infomatek.v26i2.19255.

[24] A. Almadira, Y. Pratama, and F. Purwani, "Melindungi Data Di Dunia Digital: Peran Stategis Enkripsi Dalam Keamanan Data," *Journal of Scientech Research and Development* , vol. 6, no. 2, pp. 540–549, 2024.

[25] M. T. Abdillah, I. Kurniastuti, F. A. Susanto, and F. Yudianto, "Implementasi Black Box Testing dan Usability Testing pada Website Sekolah MI Miftahul Ulum Warugunung Surabaya," *Journal of Computer Science and Visual Communication Design*, vol. 8, no. 1, pp. 234–242, 2023.