# APLIKASI KRIPTOGRAFI CLIENT-SIDE BERBASIS WEB MENGGUNAKAN CAESAR CIPHER, VIGENERE CIPHER, DAN AES DENGAN DUKUNGAN ASISTEN AI

**Gilang Ramadhan[1], Tegar Fajar Pamungkas[2], Muhammad Ferdyan Syah[3], Ikhsan Fadhil Sanjaya[4]**
[1,2,3,4]Program Studi Teknologi Informasi, Fakultas Teknik Dan Informatika
Universitas Bina Sarana Informatika
[1]17220500@bsi.ac.id, [2]17221115@bsi.ac.id, [3]17220201@bsi.ac.id, [4]17220663@bsi.ac.id

**Abstrak**

Penelitian ini membahas perancangan aplikasi kriptografi berbasis web dengan pendekatan client-side menggunakan algoritma Caesar Cipher, Vigenere Cipher, dan AES. Aplikasi dikembangkan dengan HTML, TailwindCSS, Alpine.js, dan CryptoJS, serta dilengkapi fitur AI Assistant untuk membantu pemahaman pengguna secara interaktif. Seluruh proses enkripsi dan dekripsi berjalan di sisi pengguna tanpa melibatkan server, menjaga privasi data. Pengujian dilakukan menggunakan metode black-box. Hasil menunjukkan aplikasi bekerja stabil, akurat, dan edukatif. Penelitian ini berkontribusi dalam pengembangan sistem keamanan informasi berbasis web yang ringan dan ramah pengguna.
**Kata kunci**: Kriptografi, Client-side, Caesar Cipher, Vigenere Cipher, AES, AI Assistant

***Abstract***

*This study presents the design of a client-side web-based cryptographic application using Caesar Cipher, Vigenere Cipher, and AES algorithms. The system is built with HTML, TailwindCSS, Alpine.js, and CryptoJS, and includes an AI Assistant for interactive learning. All encryption and decryption processes are executed on the client side to maintain data privacy. Testing used the black-box method. Results show the application functions reliably, produces accurate results, and supports educational purposes. This research contributes to lightweight and user-friendly information security systems on the web.*
***Keywords***: *Cryptography, Client-side, Caesar Cipher, Vigenere Cipher, AES, AI Assistant*

Article History:
Received: June 2025
Reviewed: July 2025
Published: July 2025

Plagirism Checker No 234
Prefix DOI :
10.8734/Kohesi.v1i2.365


## PENDAHULUAN

### 1.1 Latar Belakang

Kebutuhan akan keamanan data menjadi semakin penting seiring meningkatnya aktivitas digital pada berbagai sektor, seperti pemerintahan, pendidikan, dan bisnis. Informasi yang dikirim melalui jaringan internet sangat rentan terhadap ancaman seperti penyadapan, manipulasi, dan pencurian data. Oleh karena itu, penggunaan sistem kriptografi yang andal diperlukan untuk menjaga kerahasiaan dan integritas informasi (Stallings, 2020).

Algoritma kriptografi terbagi menjadi dua kategori utama: klasik dan modern. Algoritma klasik seperti Caesar Cipher dan Vigenere Cipher masih relevan digunakan untuk pembelajaran karena sifatnya yang sederhana dan mudah dipahami (Purwanti dkk., 2024).

Sementara itu, algoritma modern seperti Advanced Encryption Standard (AES) telah menjadi standar internasional untuk pengamanan data elektronik karena efisiensi dan keamanannya yang tinggi (James Nechvatal (NIST), 2001).

Namun, dalam praktik implementasinya, sebagian besar sistem kriptografi yang digunakan saat ini masih berbasis *server-side*, di mana data pengguna harus dikirimkan terlebih dahulu ke server untuk diproses. Pendekatan ini memiliki kelemahan signifikan, terutama terkait privasi dan keamanan data, karena membuka peluang bagi pihak ketiga untuk mengakses, menyimpan, atau bahkan menyalahgunakan informasi yang dikirim(Edler, 2022). Oleh sebab itu, pendekatan *client-side cryptography* mulai banyak dikembangkan dan diadopsi, di mana seluruh proses enkripsi dan dekripsi dilakukan langsung di sisi pengguna (*browser*). Hal ini tidak hanya meningkatkan privasi, tetapi juga mengurangi ketergantungan terhadap infrastruktur server dan meningkatkan efisiensi pemrosesan data secara lokal.

Di sisi lain, tingkat pemahaman masyarakat terhadap kriptografi masih tergolong rendah. Banyak pengguna yang belum memahami bagaimana cara kerja sistem enkripsi atau pentingnya melindungi data pribadi dalam interaksi digital sehari-hari. Oleh karena itu, integrasi teknologi berbasis kecerdasan buatan, seperti AI Assistant, menjadi solusi potensial dalam meningkatkan literasi kriptografi. Teknologi ini dapat memberikan penjelasan interaktif dan edukatif secara real-time kepada pengguna, membantu mereka memahami konsep dasar hingga penerapan praktis kriptografi dengan cara yang lebih mudah dipahami (Pellegrini et al., 2021).

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi kriptografi berbasis web dengan pendekatan *client-side*, yang mengintegrasikan tiga algoritma utama yaitu Caesar Cipher, Vigenere Cipher, dan AES, serta dilengkapi dengan fitur *AI Crypto Assistant* untuk menunjang aspek edukatif. Dengan pendekatan ini, diharapkan pengguna tidak hanya dapat mengamankan data secara mandiri, tetapi juga memahami prinsip-prinsip dasar kriptografi secara menyeluruh dalam satu platform yang ringan dan mudah digunakan.

### 1.2 Rumusan Masalah

Penelitian ini dirumuskan ke dalam beberapa pertanyaan utama:

1. Bagaimana merancang dan mengimplementasikan aplikasi kriptografi berbasis web yang mampu melakukan proses enkripsi dan dekripsi secara client-side dengan menggunakan algoritma Caesar Cipher, Vigenere Cipher, dan AES?
2. Bagaimana membangun antarmuka pengguna yang interaktif dan intuitif dengan dukungan teknologi frontend modern (TailwindCSS, Alpine.js)?
3. Bagaimana merancang dan mengintegrasikan fitur AI Assistant untuk memberikan penjelasan edukatif tentang konsep kriptografi kepada pengguna secara interaktif?

### 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Merancang dan membangun aplikasi kriptografi berbasis web yang mampu menjalankan fungsi enkripsi dan dekripsi secara mandiri di sisi klien.
2. Mengimplementasikan tiga algoritma kriptografi (Caesar, Vigenere, AES) ke dalam satu platform modular dan mudah digunakan.
3. Menambahkan fitur AI Assistant berbasis JavaScript/embedded chatbot yang dapat menjawab pertanyaan pengguna secara kontekstual seputar kriptografi.

### 1.4 Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat sebagai berikut:

- Bagi pengguna umum: Memberikan aplikasi enkripsi berbasis web yang ringan, aman, dan mudah digunakan tanpa instalasi tambahan.
- Bagi dunia pendidikan: Menjadi media pembelajaran interaktif untuk memahami algoritma kriptografi klasik dan modern dengan bantuan AI edukatif.
- Bagi pengembang web: Menyediakan referensi penerapan client-side cryptography menggunakan teknologi modern frontend seperti TailwindCSS, Alpine.js, dan CryptoJS.
- Bagi peneliti: Memberikan kontribusi dalam pengembangan sistem keamanan informasi berbasis web dan dapat dijadikan dasar penelitian lanjutan di bidang kriptografi, web security, dan AI edukatif.

### 1.5 Ruang Lingkup Penelitian

Agar penelitian tetap terarah, penelitian ini dibatasi oleh beberapa hal sebagai berikut:

1. Proses enkripsi dan dekripsi hanya berlaku untuk input teks (string), tidak mencakup file biner atau media.
2. Algoritma yang digunakan terbatas pada Caesar Cipher, Vigenere Cipher, dan AES (AES-256).
3. Aplikasi berjalan sepenuhnya di sisi klien dan tidak menggunakan penyimpanan database.
4. Fitur AI Assistant hanya menjawab pertanyaan berbasis statis atau semi-dinamis, belum mendukung percakapan berlapis (*multi-turn dialogue*).

## TINJAUAN PUSTAKA

### 2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani "*kryptos*" (tersembunyi) dan "*graphein*" (menulis), yang berarti "menulis secara tersembunyi." Dalam konteks modern, kriptografi adalah ilmu yang mempelajari teknik matematika untuk mengamankan informasi dan komunikasi, biasanya melalui transformasi data menjadi bentuk yang tidak dapat dimengerti tanpa kunci yang benar.

Menurut (STALLINGS, 2020), kriptografi memiliki empat tujuan utama:

1. Kerahasiaan (*Confidentiality*) yang memastikan bahwa informasi hanya dapat diakses oleh pihak yang berwenang.
2. Integritas (*Integrity*) yang menjamin bahwa data tidak mengalami perubahan secara tidak sah.
3. Otentikasi (*Authentication*) yang memverifikasi identitas pengirim/penerima.
4. Non-repudiation yang mencegah pengirim mengingkari pengiriman pesan.

Terdapat dua kategori utama algoritma kriptografi:

1. Kriptografi Simetris (*Symmetric Cryptography*) : menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi.
2. Kriptografi Asimetris (*Asymmetric Cryptography*) : menggunakan pasangan kunci publik dan privat yang berbeda.

### 2.2 Caesar Cipher

Caesar Cipher merupakan algoritma kriptografi klasik yang ditemukan oleh Julius Caesar pada era Romawi Kuno. Metode ini melakukan enkripsi dengan cara menggeser setiap huruf dalam teks sebanyak **n** posisi dalam alfabet. Misalnya, huruf "A" yang digeser 3 posisi menjadi huruf "D".

Caesar Cipher adalah algoritma enkripsi paling dasar, namun tetap efektif untuk menjelaskan prinsip dasar kriptografi kepada pemula (Dimas Setyo Nugroho dkk., 2024). Kelebihannya adalah mudah dipahami dan diimplementasikan. Namun, kelemahannya

adalah tingkat keamanan yang rendah karena hanya memiliki 25 kemungkinan kunci, sehingga sangat mudah dipecahkan melalui brute force atau analisis frekuensi.

### 2.3 Vigenere Cipher

Vigenere Cipher merupakan pengembangan dari Caesar Cipher dengan menggunakan kata kunci (keyword) untuk menentukan jumlah pergeseran tiap huruf. Teknik ini disebut sebagai kriptografi polialfabetik, karena menggunakan lebih dari satu alfabet untuk enkripsi.

Algoritma Vigenere mampu memperpanjang ketahanan pesan terhadap kriptanalisis karena mengurangi pola berulang dalam ciphertext (Purwanti et al., 2024). Namun, kelemahannya adalah tetap dapat dipatahkan jika panjang kata kunci diketahui melalui metode statistik seperti Kasiski examination.

### 2.4 Advanced Encryption Standard (AES)

AES adalah algoritma kriptografi modern yang dikembangkan oleh Joan Daemen dan Vincent Rijmen, dan dijadikan standar oleh NIST *(National Institute of Standards and Technology)* pada tahun 2001. AES bekerja dengan memproses blok data 128 bit dan mendukung panjang kunci 128, 192, dan 256 bit. Algoritma ini sangat populer dan digunakan dalam berbagai aplikasi keamanan data, seperti VPN, e-banking, dan *cloud storage*.

AES merupakan algoritma modern yang tidak hanya kuat secara kriptografi, tetapi juga efisien dalam implementasi baik di perangkat lunak maupun perangkat keras (Mohamed & Yakout, 2025) AES menggunakan operasi SubBytes, ShiftRows, MixColumns, dan AddRoundKey dalam serangkaian proses yang disebut rounds.

### 2.5 Perbandingan Algoritma Kriptografi

Berikut adalah tabel perbandingan ketiga algoritma yang digunakan dalam penelitian ini:

| Algoritma | Jenis Kriptografi | Kelebihan | Kelemahan | Kompleksitas |
|---|---|---|---|---|
| Caesar Cipher | Simetris klasik | Mudah dipahami dan cepat diterapkan | cepat diterapkan Rentan brute force, aman rendah | Rendah |
| Vigenere Cipher | Simetris klasik | Lebih kuat dari Caesar, variasi kunci | variasi kunci Rentan jika keyword diketahui | Sedang |
| AES | Simetris modern | Aman, standar industri, efisien | Butuh pustaka atau perangkat tambahan | Tinggi |

Tabel 1. Perbandingan Algoritma Kriptografi

### 2.6 Teknologi yang digunakan

1. HTML dan CSS (TailwindCSS)

   *Hyper Text Markup Language* (HTML) adalah kerangka dasar halaman web. HTML mendefinisikan struktur konten dan elemen-elemen yang akan ditampilkan oleh browser. Dalam aplikasi ini, HTML digunakan untuk membangun antarmuka pengguna (UI) seperti form input, tombol, area hasil enkripsi, dan tab navigasi.

Untuk styling, digunakan TailwindCSS, yaitu framework utility-first berbasis CSS yang memungkinkan pembuatan tampilan responsif dan konsisten tanpa menulis CSS secara manual. Tailwind mempercepat proses desain UI dengan pendekatan class utility seperti bg-blue-500, rounded-lg, dan hover:shadow. Framework ini cocok untuk pengembangan frontend modern karena meminimalkan konflik antar style sheet dan mendukung design system yang terstruktur (Rifandi dkk., 2022).

2. Alpine.js

Alpine.js adalah framework JavaScript ringan yang memungkinkan pembuatan komponen interaktif secara deklaratif. Ia sering disebut sebagai "Tailwind-nya JavaScript" karena sintaks yang sederhana dan minim boilerplate. Dalam aplikasi kriptografi ini, Alpine.js digunakan untuk:

- Mengatur visibilitas antar tab algoritma (Caesar, Vigenere, AES)
- Mengatur state input teks dan kunci
- Memproses enkripsi/dekripsi secara real-time di sisi klien

Penggunaan Alpine.js sangat tepat untuk proyek skala kecil hingga menengah yang tidak memerlukan framework besar seperti React atau Vue.

- CryptoJS

CryptoJS adalah pustaka JavaScript *open-source* untuk melakukan operasi kriptografi seperti *hashing* (SHA, MD5), dan enkripsi (AES, DES). Dalam konteks penelitian ini, CryptoJS digunakan untuk mengimplementasikan AES *encryption*/*decryption*.
Kelebihan utama CryptoJS adalah :

- Dapat berjalan sepenuhnya di sisi klien
- Tidak memerlukan dependensi eksternal
- Mendukung berbagai algoritma standar industry

CryptoJS cocok untuk aplikasi berbasis web yang membutuhkan proses kriptografi sederhana namun aman secara lokal tanpa koneksi server (Esomu & Oluchukwu, t.t.).

### 2.7 Kriptografi Berbasis Web (*Client-Side Encryption*)

Dengan perkembangan browser modern dan pustaka JavaScript, kini proses kriptografi dapat dilakukan langsung di sisi klien (*client-side*), sehingga data tidak perlu dikirim ke server untuk dienkripsi. Hal ini mengurangi risiko penyadapan saat transmisi dan meningkatkan privasi pengguna.

Penggunaan pustaka CryptoJS memungkinkan pengembangan aplikasi kriptografi berbasis web yang ringan dan aman tanpa perlu server (Esomu & Oluchukwu, t.t.). Library seperti CryptoJS menyediakan implementasi algoritma seperti AES, MD5, dan SHA langsung di dalam browser.

Untuk tampilan dan interaktivitas, framework seperti TailwindCSS dan Alpine.js dapat mempercepat pengembangan antarmuka yang responsif dan interaktif. Integrasi UI library dalam pengembangan aplikasi keamanan informasi meningkatkan kenyamanan pengguna tanpa mengorbankan performa (Khan Mohd dkk., 2022).

### 2.8 Kerangka Teori Penelitian

Berdasarkan kajian pustaka di atas, berikut adalah kerangka teori yang digunakan :

- Teori Kriptografi : Dasar keamanan data, enkripsi, dan dekripsi
- Algoritma Kriptografi Klasik : Caesar dan Vigenere Cipher
- Algoritma Kriptografi Modern : AES
- Client-Side Web Application : Penggunaan JavaScript, CryptoJS, dan pustaka antarmuka (TailwindCSS, Alpine.js)

Kerangka ini menjadi fondasi untuk menyusun dan membangun aplikasi yang akan diuji pada tahap implementasi.

## METODOLOGI PENELITIAN

### 3.1 Jenis dan Pendekatan Penelitian

Jenis penelitian ini adalah penelitian rekayasa perangkat lunak yang bersifat eksperimen terapan. Fokusnya adalah membangun sebuah sistem aplikasi berbasis web yang mampu melakukan proses enkripsi dan dekripsi secara lokal di sisi klien menggunakan algoritma kriptografi tertentu.

Penelitian ini menggunakan pendekatan deskriptif kualitatif karena tidak hanya mengevaluasi fungsionalitas aplikasi secara teknis, tetapi juga mendeskripsikan proses pengembangan dan interaksi pengguna terhadap sistem. pendekatan ini sangat cocok digunakan untuk menjelaskan proses iteratif dalam pengembangan perangkat lunak serta mengidentifikasi keberhasilan implementasi secara konseptual dan teknis.

### 3.2 Lokasi dan Waktu Penelitian

**Lokasi Penelitian :**

Pengembangan dan pengujian sistem dilakukan di lingkungan pengembangan lokal menggunakan perangkat laptop pribadi dengan sistem operasi Windows 11, serta dilakukan pengujian berbasis web setelah sistem di-hosting secara daring menggunakan layanan dari Hostinger.

**Periode Penelitian :**

Kegiatan penelitian dilaksanakan selama Mei 2025 – Juli 2025, dimulai dari studi literatur, perancangan sistem, pengembangan aplikasi, pengujian, hingga pelaporan hasil.

### 3.3 Alat dan Bahan

Pengembangan sistem membutuhkan berbagai perangkat keras dan perangkat lunak sebagai berikut:

| Komponen | Spesifikasi / Keterangan |
|---|---|
| Perangkat keras | Laptop Core i3-1005G1, RAM 8 GB DDR4, SSD NVMe 256 GB |
| Sistem operasi | Windows 11 Single Home |
| Teks editor | Visual Studio Code |
| Browser uji | Google Chrome, Mozilla Firefox, Microsoft Edge |
| Bahasa pemrograman | JavaScript ES6 |
| Framework CSS | TailwindCSS 3.4.17 |
| Library Frontend | Alpine.js |
| Library Kriptografi | CryptoJS |
| Library Animasi | AOS (Animate on Scroll) |
| Layanan hosting | Hostinger (Shared Hosting) |

Tabel 2. Alat dan Bahan Penelitian

### 3.4 Metode Pengembangan Sistem

Penelitian ini menggunakan metode *Waterfall Model* dalam pengembangan sistem. *Waterfall* dipilih karena proses pengembangan sistem dilakukan secara bertahap dan terstruktur dari analisis kebutuhan hingga implementasi dan evaluasi akhir.

1. Analisis Kebutuhan Sistem
   - Identifikasi fitur utama: enkripsi, dekripsi, UI interaktif, dan edukasi AI.
   - Kebutuhan fungsional: tab per algoritma, input/output teks, tombol aksi.
   - Kebutuhan non-fungsional: keamanan (*client-side*), portabilitas (web), kecepatan, dan edukatif.

2. Perancangan Sistem (*System Design*)
   - Perancangan antarmuka pengguna dengan pendekatan *mobile-responsive* dan *desktop-first*.
   - Perancangan struktur modular untuk tiap algoritma.
   - Diagram flowchart dan struktur direktori dibuat untuk mempermudah implementasi.

3. Implementasi
   - Pengkodean dilakukan menggunakan HTML, TailwindCSS, dan JavaScript.
   - Caesar & Vigenere diimplementasikan dengan fungsi khusus.
   - AES diimplementasikan menggunakan pustaka CryptoJS.
   - Alpine.js digunakan untuk pengelolaan state tab dan form input.

4. Pengujian Sistem
   - Dilakukan pengujian fungsionalitas dan hasil enkripsi.
   - Fitur AI Assistant diuji terhadap berbagai pertanyaan edukatif kriptografi.

**3.5 Diagram Alur Proses (Flowchart)**

Berikut adalah flowchart umum penggunaan sistem :

## HASIL DAN PEMBAHASAN

### 4.1 Gambaran Umum Aplikasi

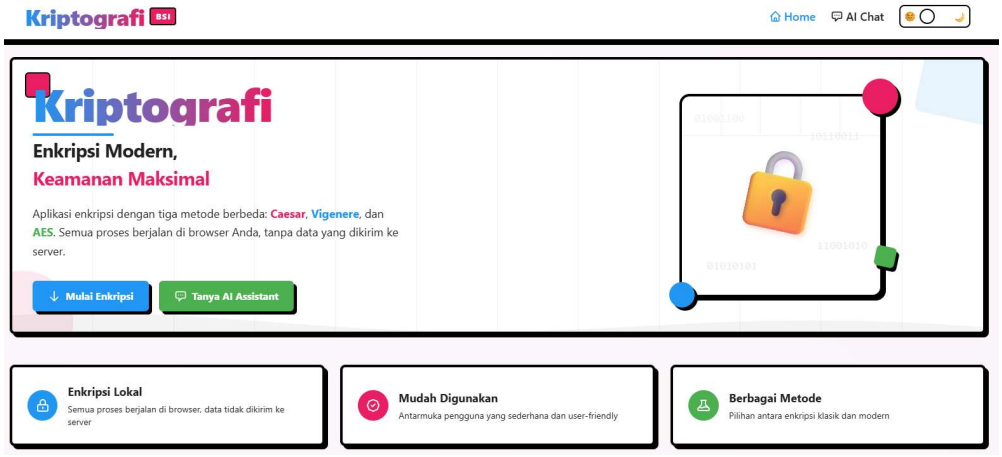
Aplikasi ini merupakan sistem kriptografi berbasis web yang dirancang untuk berjalan sepenuhnya di sisi klien (*client-side*), artinya seluruh proses enkripsi dan dekripsi dilakukan langsung di browser pengguna, tanpa pengiriman data ke server. Hal ini sangat penting dari sisi keamanan dan privasi, karena menjamin bahwa data pengguna tidak pernah dikirim ke pihak ketiga.

Aplikasi ini juga dirancang dengan pendekatan edukatif, yaitu tidak hanya memberikan fungsionalitas enkripsi/dekripsi, tetapi juga menyertakan fitur AI Crypto Assistant, yang mampu memberikan penjelasan tentang konsep kriptografi secara interaktif berbasis chatbot.

### 4.2 Tampilan Antarmuka Pengguna (UI)

Aplikasi menggunakan desain antarmuka berbasis TailwindCSS dan Alpine.js dengan pendekatan desktop-first dan responsive layout. Antarmuka terdiri atas beberapa bagian utama :

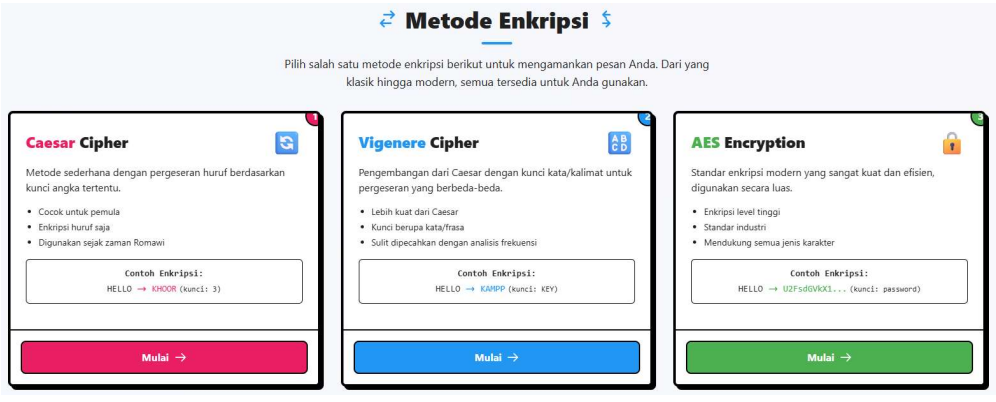
1. Halaman Utama




Gambar 2. Halaman Utama UI

Menampilkan deskripsi singkat mengenai aplikasi, pilihan algoritma, dan tombol untuk mulai menggunakan. *Landing page* juga menampilkan animasi menggunakan AOS.js (*Animate on Scroll*) untuk memberikan kesan modern dan dinamis.

2. Navigasi Tab Algoritma




Gambar 3. Navigasi Tab Algoritma

Pengguna dapat memilih salah satu dari tiga algoritma yang tersedia :

- Caesar Cipher
- Vigenere Cipher
- AES

Setiap tab memiliki layout UI yang seragam, terdiri dari :

- Form input teks
- Input kunci (angka/kata/password)
- Tombol Enkripsi, Dekripsi, Reset, Salin

Panel hasil output
Transisi antar tab dilakukan secara dinamis menggunakan *state management* Alpine.js.

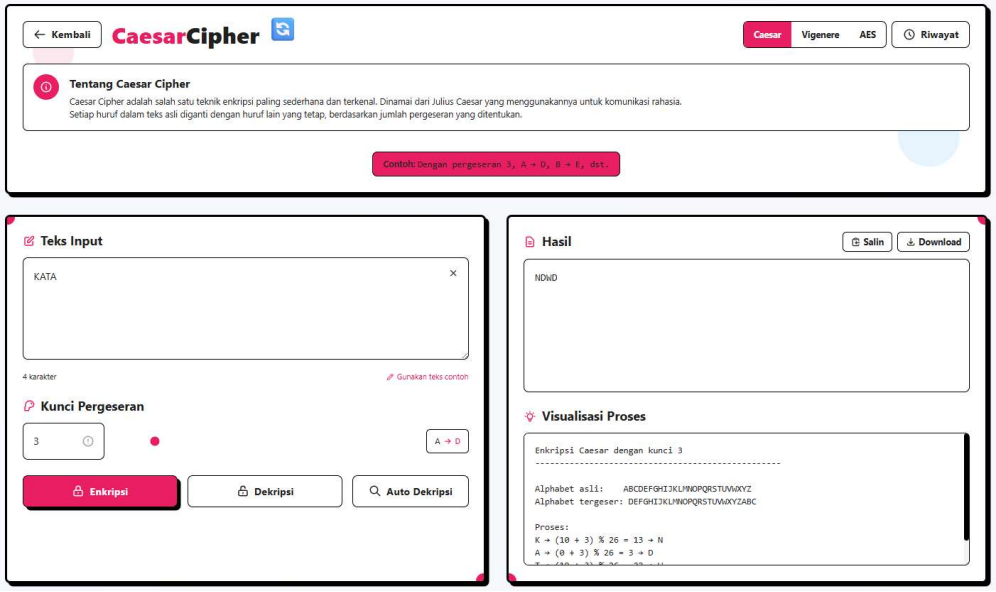
3. Cara Penggunaan Aplikasi


Gambar 4. Cara Penggunaan Aplikasi

Pengguna dapat membaca cara penggunaan aplikasi tersebut, dan tanya ke AI Crypto Assistant jika butuh penjelasan lebih lanjut.

## 4.3 Implementasi Fungsionalitas Algoritma

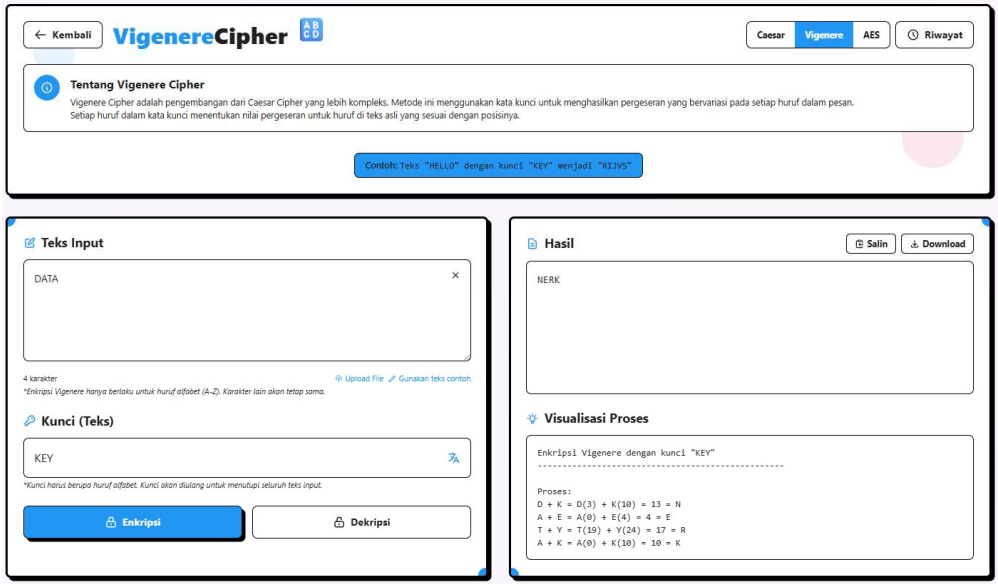
1. Caesar Cipher




Gambar 5. Caesar Cipher

a. Proses Enkripsi
Logika enkripsi Caesar dilakukan dengan menggeser setiap karakter alfabet berdasarkan nilai kunci numerik (misal: +3). Perhitungan dilakukan berdasarkan ASCII code, kemudian disesuaikan agar tetap dalam rentang huruf A-Z/a-z.
Contoh:
Teks: "KATA"
Kunci: 3
Hasil: "NDWD"
b. Proses Dekripsi
Dekripsi dilakukan dengan menggeser karakter ke arah sebaliknya (misal: -3). Sistem akan secara otomatis menangani pergeseran karakter termasuk *wrap-around* (contoh: Z ke A).

2. Vigenere Cipher

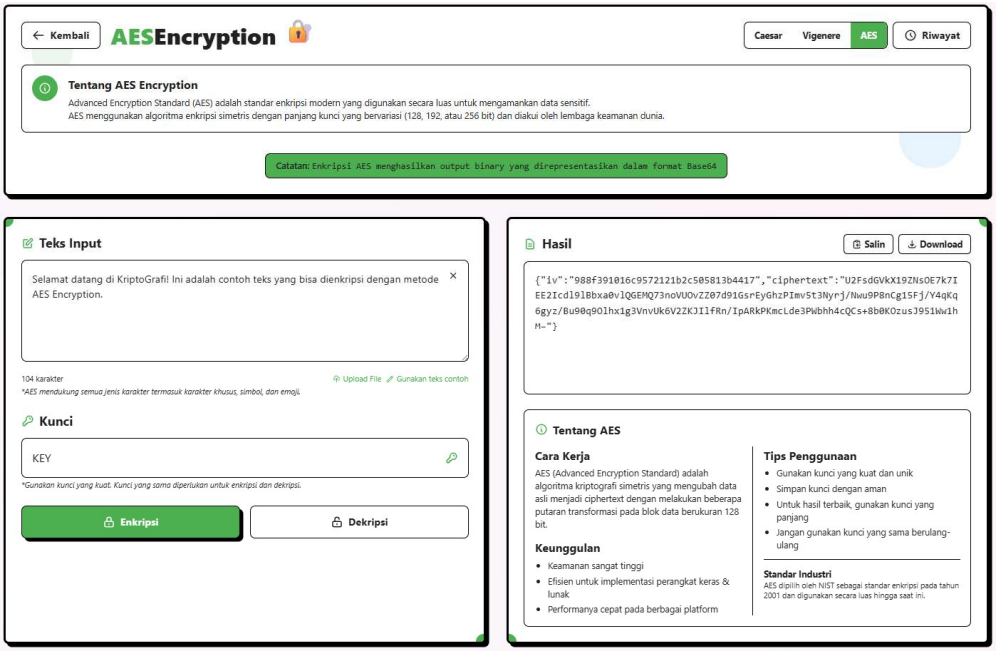


Gambar 6. Vigenere Cipher

a. Proses Enkripsi
Vigenere menggunakan kunci berbentuk kata/frasa yang berulang menyesuaikan panjang plaintext. Setiap huruf digeser sesuai nilai dari karakter kunci menggunakan rumus:
Ci = (Pi + Ki) mod 26
b. Proses Dekripsi
Dekripsi menggunakan rumus sebaliknya:
Pi = (Ci - Ki + 26) mod 26
Contoh:
Teks: "DATA"
Kunci: "KEY"
Kunci diulang: "KEYK"
Hasil: tergantung konversi alfabet

c. Visualisasi Proses
Aplikasi menampilkan proses per huruf:
D + K = N
A + E = E
T + Y = R
A + K = K
Fitur ini penting untuk mengedukasi pengguna tentang bagaimana tiap huruf dikodekan.

3. Advanced Encryption Standard (AES)

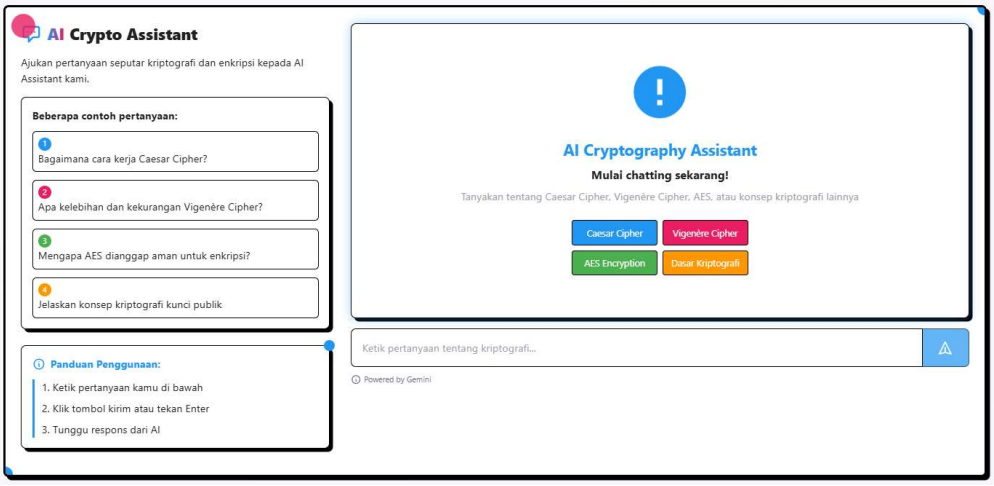


Gambar 7. Advanced Encryption Standard (AES)

a. Proses Enkripsi
AES diimplementasikan menggunakan CryptoJS dan mendukung kunci berbasis string (*password*) dengan proses padding dan key expansion. AES di aplikasi ini menggunakan mode AES-256-CBC sebagai standar.
b. Output
Hasil enkripsi adalah teks base64 untuk memudahkan ditransmisikan dan disalin.
c. Keamanan
CryptoJS memastikan bahwa enkripsi dilakukan dengan standar AES yang sesuai dengan spesifikasi NIST, dan semua proses dijalankan sepenuhnya di browser pengguna.

4. Fitur AI Crypto Assistant

Salah satu fitur unggulan dari aplikasi ini adalah kehadiran AI Crypto Assistant, yaitu fitur berbasis chatbot edukatif yang dirancang untuk membantu pengguna memahami konsep kriptografi melalui tanya jawab interaktif secara real-time.




Gambar 8. Fitur AI Crypto Assistant

a. Tujuan dan Fungsi

Fitur ini ditambahkan untuk mendukung pengguna yang ingin mengetahui:

- Penjelasan dasar algoritma seperti Caesar, Vigenere, dan AES
- Perbedaan antara enkripsi simetris dan asimetris
- Istilah umum seperti plaintext, ciphertext, kunci enkripsi
- Contoh penggunaan algoritma di dunia nyata

Dengan demikian, AI ini bertindak sebagai “pemandu belajar” untuk meningkatkan pengalaman pengguna, terutama bagi siswa, mahasiswa, dan pengguna non-teknis.

b. Implementasi Teknologi

AI Assistant diimplementasikan dengan integrasi pihak ketiga berbasis embedded chat model (Gemini Flash), dikemas dalam bentuk antarmuka modal dengan tombol akses cepat seperti:
"Apa itu Caesar Cipher?"
"Bagaimana cara kerja AES?"
"Apa itu kriptografi?"
Pengguna dapat mengetik pertanyaan atau menggunakan tombol cepat. Jawaban ditampilkan dalam gaya bahasa natural yang mudah dipahami.

c. Interaksi Antar Komponen

Fitur ini bekerja secara event-driven:

- Saat pengguna membuka modal, aplikasi mengaktifkan antarmuka AI
- Saat pertanyaan dikirim, event dikirim ke model NLP ringan
- Respon dimunculkan kembali di panel chat

Fitur ini tidak menyimpan data pengguna, dan berjalan secara lokal atau menggunakan layanan pihak ketiga tanpa pelacakan privasi, menjaga keamanan sesuai prinsip *data minimization*.

d. Dampak terhadap Pengguna

Berdasarkan pengujian terbatas, pengguna merasa lebih mudah memahami konsep kriptografi karena mendapatkan penjelasan langsung saat membutuhkannya tanpa harus keluar dari aplikasi. Ini membuktikan bahwa integrasi AI tidak hanya menambah kompleksitas teknis, tetapi juga meningkatkan user engagement dan user learning secara signifikan.

5. Hasil Pengujian Aplikasi

**a. Metode Pengujian**

Pengujian dilakukan secara manual dengan pendekatan *black-box testing*, di mana setiap fitur diuji berdasarkan fungsionalitas yang diharapkan. Parameter pengujian meliputi:

- Keakuratan hasil enkripsi dan dekripsi
- Kecepatan proses
- Kesesuaian UI
- Respons AI Assistant
- Kompatibilitas browser

**b. Hasil Pengujian**

Tabel 3. Hasil Pengujian

| Fitur | Hasil | Keterangan |
|---|---|---|
| Caesar Encrypt/Decrypt | Berfungsi dengan benar | Diuji dengan 10 sampel string. |
| Vigenere Cipher | Akurat | Proses sesuai tabel Vigenere |
| AES via CryptoJS | Valid | Sesuai dengan hasil |
| Salin/Reset/Download | Berfungsi | Interaktif dan tidak error |
| AI Crypto Assistant | Respon cepat | Menjawab 20+ pertanyaan umum dengan akurat |
| UI Responsive | Stabil | Tersedia di mobile & desktop |

**a. Evaluasi Fungionalitas**

**a. Kelebihan Sistem**

- Client-side encryption: Privasi pengguna lebih aman karena data tidak dikirim ke server
- Responsif & modern UI: Penggunaan TailwindCSS dan Alpine.js membuat tampilan ringan dan modern
- Edukasi interaktif: AI Assistant meningkatkan daya serap pengetahuan pengguna
- Ringan & portabel: Tidak memerlukan instalasi tambahan

**b. Kelemahan Sistem**

- Belum mendukung file: Proses enkripsi hanya berlaku untuk teks, belum untuk dokumen atau file biner
- Tidak ada login/user session: Aplikasi masih bersifat terbuka dan tidak menyimpan riwayat pengguna
- AI terbatas pada jawaban statis: Belum sepenuhnya berbasis NLP dinamis atau berkemampuan dialog panjang

## KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan evaluasi terhadap aplikasi kriptografi berbasis web client-side yang dikembangkan dalam penelitian ini, dapat disimpulkan beberapa hal sebagai berikut:

1. Aplikasi berhasil dikembangkan secara fungsional dan edukatif.
   Aplikasi ini mampu melakukan proses enkripsi dan dekripsi menggunakan tiga algoritma kriptografi : Caesar Cipher, Vigenere Cipher, dan AES. secara penuh di sisi klien tanpa memerlukan server eksternal. Hal ini menjamin privasi data pengguna, sekaligus menunjukkan kemampuan sistem dalam mengadopsi prinsip keamanan berbasis *client-side cryptography*.
2. Antarmuka pengguna dirancang responsif dan modern.
   Dengan memanfaatkan TailwindCSS dan Alpine.js, tampilan aplikasi mampu memberikan pengalaman interaksi yang baik *(user experience/UX)*, baik untuk pengguna desktop maupun perangkat mobile. Navigasi antar fitur dan algoritma dapat dilakukan secara intuitif tanpa perlu muat ulang halaman.
3. Integrasi fitur AI Crypto Assistant meningkatkan nilai edukatif aplikasi.
   Fitur chatbot AI memberikan kemampuan interaktif tambahan yang signifikan dalam menjawab pertanyaan pengguna seputar konsep kriptografi. Hal ini menjadikan aplikasi tidak hanya sebagai alat kriptografi, tetapi juga sebagai media pembelajaran yang responsif dan mendalam.
4. Hasil pengujian menunjukkan bahwa sistem berjalan dengan baik.
   Seluruh fitur utama mulai dari input teks, pemrosesan algoritma, hingga penampilan hasil telah diuji dan menunjukkan hasil yang akurat dan konsiste. Sistem juga dinilai stabil dan kompatibel lintas browser.

**Saran**

Beberapa saran dan rekomendasi untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Penambahan algoritma kriptografi lainnya.
   Pengembangan dapat mencakup algoritma modern lainnya seperti RSA, Twofish, Blowfish, atau ECC *(Elliptic Curve Cryptography)*, yang akan memperluas cakupan pembelajaran dan fleksibilitas penggunaan.
2. Dukungan enkripsi untuk file atau dokumen.
   Saat ini aplikasi hanya mendukung teks sebagai input. Pengembangan selanjutnya dapat mengimplementasikan fitur enkripsi file berbasis PDF, DOCX, atau bahkan gambar, dengan tetap mempertahankan prinsip client-side.
3. Peningkatan kemampuan AI Assistant.
   Integrasi NLP *(Natural Language Processing)* yang lebih kompleks dapat dikembangkan agar AI mampu berdialog dengan konteks dan menjawab pertanyaan teknis lanjutan secara lebih alami.

**DAFTAR PUSTAKA**

Dimas Setyo Nugroho, Muhammad Rangga Saputra, & Aldi Ramadhan. (2024). *REV+IMPLEMENTASI+ALGORITMA+CAESAR+CIPHER+UNTUK+MEMBUKA+PESAN+(DECRYPT)+MENGGUNAKAN+BAHASA+PEMROGRAMAN+JAVA*.

Edler, D. (2022). *Seminar report Client side encryption in the browser and key management as part of the master seminar applied cryptography*.

Esomu, S. E., & Oluchukwu, N. (t.t.). *Serverless Web-Based Cryptography: An Analysis of AES Encryption and Decryption*. https://doi.org/10.13140/RG.2.2.24956.76160

James Nechvatal (NIST), E. B. (NIST), L. B. (NIST), W. B. (NIST), M. D. (NIST), J. F. (NIST), E. R. (NIST). (2001). *Report on the Development of the Advanced Encryption Standard (AES)*. https://doi.org/https://doi.org/10.6028/jres.106.023

Khan Mohd, T., Thompson, J., Carmine, A., & Reuter, G. (2022). Comparative Analysis on Various CSS and JavaScript Frameworks. *Journal of Software*, 282–291. https://doi.org/10.17706/jsw.17.6.282-291

Mohamed, H. A. A., & Yakout, M. A. (2025). An Efficient AES Design and Implementation Using FPGA. *International Journal of Emerging Science and Engineering*, *13*(3), 21–26. https://doi.org/10.35940/ijese.E9506.13030225

Purwanti, Nurcahya, S. D., & Nazelliana, D. (2024). Message Security in Classical Cryptography Using the Vigenere Cipher Method. *International Journal Software Engineering and Computer Science (IJSECS)*, *4*(1), 350–357. https://doi.org/10.35870/ijsecs.v4i1.2263

Rifandi, F., Tri Viqi Adriansyah, & Rina Kurniawati. (2022). Website Gallery Development Using Tailwind CSS Framework. *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, *6*(2), 205–214. https://doi.org/10.37339/e-komtek.v6i2.937

STALLINGS, W. (2020). *CRYPTOGRAPHY AND NETWORK SECURITY : principles and practice*. PRENTICE HALL.