Jurnal Informatika dan Sistem Informasi (JIFoSI)
Vol. 6, No. 2. Agustus 2025
e-ISSN : 2722-130
DOI: https://doi.org/10.33005/jifosi.v6i2.483

IMPLEMENTASI KRIPTOGRAFI UNTUK MELINDUNGI INFORMASI TRANSAKSI PADA *E-COMMERCE* MENGGUNAKAN METODE *CAESAR CIPHER* DAN *RC4*

Paulus Vidorosa Pakan[1*)], Hari Soetanto[2)]
***corresponding author**
E-mail : [1)]paulusv.pakan@gmail.com i , [2)]hari.soetanto@budiluhur.ac.id
[1, 2]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

(Naskah masuk: 31 Desember 2024, diterima untuk diterbitkan: 31 Agustus 2025)

Abstrak

Perkembangan teknologi informasi dan komunikasi telah menyebabkan peningkatan signifikan dalam jumlah transaksi yang dilakukan melalui *website e-commerce*, menawarkan kemudahan dalam berbelanja namun juga meningkatkan risiko keamanan data. Kejadian seperti kebocoran data besar-besaran di beberapa *platform e-commerce* terkemuka telah menyoroti kebutuhan mendesak untuk perlindungan data yang lebih efektif. Dalam upaya meningkatkan keamanan ini, penelitian ini mengimplementasikan kriptografi dengan menggunakan metode *Caesar Cipher* dan *RC4*, diuji melalui enkripsi dan dekripsi data bukti bayar pengguna dalam *format JPEG, JPG, PNG, DOCX, XLSX,* dan *PDF*. Proses enkripsi rata-rata membutuhkan waktu sekitar 0,025364685 detik, sementara dekripsi membutuhkan waktu 0,036693764 detik. Tingkat keberhasilan proses enkripsi dan dekprisi adalah 100% karena selama uji coba belum pernah gagal. Kedua metode tersebut terbukti efisien dalam menjaga kerahasiaan, integritas, dan otentikasi informasi transaksi yang dipertukarkan, memberikan solusi yang tidak hanya meningkatkan keamanan tetapi juga efisiensi dalam operasional *e-commerce*. Implementasi dari *Caesar Cipher* dan *RC4* menawarkan perlindungan yang *robust* terhadap penyadapan, pencurian data, dan manipulasi oleh pihak yang tidak berwenang, mengarah pada peningkatan kepercayaan pengguna dan adopsi yang lebih luas dari layanan *e-commerce*.

Kata kunci: *Kriptografi, E-commerce, RC4, Caesar Cipher*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah mengubah cara kita melakukan berbagai aktivitas, termasuk transaksi bisnis melalui website *e-commerce. E-commerce* telah menjadi bagian integral dari kehidupan sehari-hari, memungkinkan transaksi jual beli dilakukan secara cepat dan efisien. Namun, dengan meningkatnya penggunaan *e-commerce*, muncul juga berbagai tantangan, terutama terkait dengan keamanan informasi transaksi[1].

Keamanan informasi transaksi merupakan aspek kritis dalam *e-commerce*. Setiap transaksi melibatkan pertukaran data sensitif, seperti informasi kartu kredit, detail pengiriman, dan data pribadi pelanggan. Ancaman terhadap keamanan informasi ini dapat berasal dari berbagai sumber, termasuk pencurian data, penyadapan, dan manipulasi data oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan mekanisme yang kuat untuk melindungi informasi transaksi dari ancaman tersebut[2].

Salah satu contoh kasus yang menonjol adalah kebocoran data di Tokopedia, salah satu *marketplace* terbesar di Indonesia. Pada April 2020, Tokopedia mengalami kebocoran data yang melibatkan 91 juta akun pengguna dan 7 juta akun *merchant*. Data yang bocor mencakup *email*, password, dan nama pengguna. Insiden ini menyoroti

pentingnya perlindungan data pribadi dan menunjukkan bahwa sistem keamanan yang ada masih rentan terhadap serangan siber.

Kriptografi adalah ilmu yang mempelajari teknik untuk mengamankan informasi dengan mengubahnya menjadi bentuk yang tidak dapat dibaca oleh pihak yang tidak berwenang. Dua proses utama dalam kriptografi adalah enkripsi, yang mengubah *plaintext* (teks asli) menjadi *ciphertext* (teks terenkripsi), dan dekripsi, yang mengembalikan *ciphertext* ke *plaintext.* Dalam konteks *e-commerce*, kriptografi dapat digunakan untuk melindungi data transaksi selama pengiriman dan penyimpanan[3].

Penelitian sebelumnya telah menunjukkan efektivitas berbagai metode kriptografi dalam menjaga keamanan data. Salah satu metode klasik yang sering digunakan adalah *Caesar Cipher*, yang menggantikan setiap huruf dalam plaintext dengan huruf yang berjarak tertentu dalam alfabet. Meskipun sederhana, *Caesar Cipher* dapat memberikan lapisan keamanan dasar yang cukup efektif dalam skenario tertentu[4],[5].

Algoritma *RC4*, di sisi lain, adalah salah satu algoritma kriptografi modern yang lebih kompleks dan kuat. *RC4* adalah *stream* cipher yang menggunakan *keystream* untuk mengenkripsi dan mendekripsi data. *Keystream* dihasilkan dengan menginisialisasi dan memutasi array S-box, yang kemudian di-*XOR*-kan dengan *plaintext* atau *ciphertext*[6],[7].

Karena permasalahan keamanan informasi transaksi pada website *e-commerce* yang semakin kompleks, penulis ingin menggunakan metode kriptografi *Caesar Cipher* dan *RC4* untuk melindungi informasi transaksi. Metode *Caesar Cipher* dapat digunakan untuk memberikan enkripsi dasar yang cepat dan mudah diimplementasikan, sementara *RC4* menawarkan enkripsi yang lebih kuat dan aman dengan menggunakan *keystream* yang dinamis. Implementasi kombinasi kedua metode ini diharapkan dapat menyediakan enkripsi *end- to-end* yang efektif, menjaga kerahasiaan, integritas, dan otentikasi informasi transaksi yang dipertukarkan antara pelanggan dan penyedia layanan *e-commerce*. Dengan demikian, penerapan kriptografi ini dapat meningkatkan kepercayaan dan keamanan dalam transaksi online[8].

2. METODE PENELITIAN

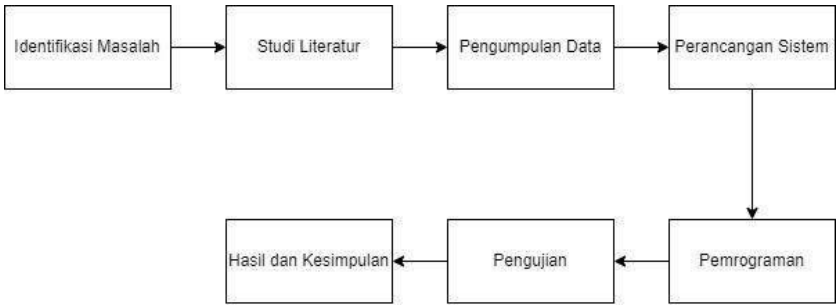
2.1 Data Penelitian

Untuk menguji sistem yang sedang dikembangkan, sangat penting menggunakan data transaksi atau bukti transaksi dari *platform e-commerce* yang telah dibuat. Data ini tidak hanya mencakup *detail* pembelian dan pembayaran tetapi juga dienkripsi dan didekripsi untuk menjamin keamanan. Penggunaan data transaksi yang terenkripsi ini memungkinkan simulasi yang mendekati skenario nyata, membantu memastikan operasional sistem yang efektif. Selain itu, sistem mendukung *format* data seperti *JPEG, PNG, JPG, GIF, dan SVG,* serta *format* dokumen seperti *docx, xlsx, dan pdf,* memperluas fleksibilitas dalam pengelolaan bukti transaksi dan meningkatkan kemampuan sistem dalam mengamankan transaksi secara efisien. Pada Gambar 1 berikut, merupakan contoh data pembayaran.

Gambar 1. Bukti Pembayaran

2.2 Metode Pengembangan Sistem

Penerapan metode adalah serangkaian kegiatan yang dilakukan secara sistematis dan terorganisir oleh peneliti untuk mencapai hasil yang diinginkan dari penelitian mereka. Pada Gambar 2 berikut, adalah prosedur penelitian dijelaskan untuk melanjutkan kegiatan selanjutnya.



Gambar 2. Metode Pengembangan Sistem

Proses ini dimulai dengan identifikasi masalah, lalu peneliti mencari dan mengumpulkan sejumlah buku, majalah, atau artikel yang berkaitan dengan masalah dan tujuan penelitian, dan diikuti oleh pengumpulan materi teoretis yang mendukung penelitian, termasuk penerapan metode *RC4* dan *Caesar Cipher*. Selanjutnya, alur penelitian melibatkan perancangan sistem yang dimulai dengan desain sistem, mencakup antarmuka pengguna, basis data, dan pemograman lanjutan untuk keamanan data. Setelah sistem dirancang, dilakukan pengujian dengan metode yang telah dikembangkan. Setelah semua tahapan ini selesai, penelitian dilanjutkan dengan analisis hasil dari metode yang telah diimplementasikan. Setelah di analisis, maka akan keluar hasil dan peneliti dapat menyimpulkannya.

2.3 Implementasi Kriptografi *Caesar Cipher* dan *RC4*

2.3.1 Algoritma Enkripsi *Caesar Chiper*

Algoritma *Caesar Cipher* dapat dijelaskan dengan menggunakan operasi modulo untuk memastikan bahwa pergeseran tetap dalam batas alfabet. Dalam konteks alfabet bahasa Inggris, yang terdiri dari 26 huruf, pergeseran dihitung dengan rumus (1) berikut[4],[6]:

$$C = (P + K) \bmod 26 \quad (1)$$

Di mana:

a. *C* adalah karakter dalam *ciphertext.*
b. *P* adalah karakter dalam *plaintext* yang telah diubah ke dalam bentuk angka (0-25).
c. *K* adalah kunci atau jumlah pergeseran.

2.3.2 Algoritma Dekripsi *Caesar Chiper*

Proses dekripsi *Caesar Cipher* adalah kebalikan dari proses enkripsi. Setiap karakter dalam *chipertext* digantikan oleh karakter yang terletak beberapa posisi ke belakang dalam alfabet. Langkah-langkahnya adalah sebagai berikut[4],[6]:

a. Tentukan jumlah pergeseran yang akan digunakan (misalnya, 3) menggunakan rumus (2).
b. Untuk setiap karakter dalam *chipertext,* gantikan dengan karakter yang berada tiga posisi di belakang dalam alfabet.

$$P = (C – K) \bmod 26 \quad (2)$$

Di mana:

a. *C* adalah karakter dalam *ciphertext.*
b. *P* adalah karakter dalam *plaintext* yang telah diubah ke dalam bentuk angka (0-25).

c. *K* adalah kunci atau jumlah pergeseran.
d. *mod 26* memastikan bahwa hasilnya tetap dalam jangkauan alfabet (0-25).

2.3.3 Algoritma *Enkripsi RC4*

Langkah-langkah algoritma enkripsi *RC4* dapat dirangkum sebagai berikut[8],[9]:

a. Tahap pertama adalah menginisialisasi array *S-box*, S[0] hingga S[255], di mana setiap elemen diisi dengan nilai 0 hingga 255 secara berurutan, artinya S[0] = 0, S[1] = 1, dan seterusnya hingga S[255] = 255.
b. Selanjutnya, array kunci *K* diatur dengan panjang yang sama, yaitu 256. Jika kunci yang diberikan lebih pendek dari 256, dilakukan pengulangan kunci tersebut hingga mencapai panjang yang diinginkan. Contoh, jika kunci awal adalah "abc", maka akan diulang menjadi "abcabcabc..." sampai terbentuk array kunci *K*[0] hingga *K*[255].
c. Langkah ketiga adalah melakukan permutasi nilai-nilai dalam array S. Nilai j diinisialisasi menjadi 0 dan untuk setiap nilai i dari 0 hingga 255, nilai j di-*update* dan elemen S[i] ditukar dengan S[j] menggunakan formula (3).

$$j = (j + S[i] + K[i]) \bmod 256$$
$$\text{Tukar } S[i] \text{ dengan } S[j] \quad (3)$$

d. Pembuatan *keystream* dilakukan dengan menginisialisasi i dan j kembali ke 0. *Keystream* dihasilkan dengan meng-*update* i dan j, menukar elemen S berdasarkan nilai i dan j yang baru, dan menggunakan nilai tersebut untuk mendapatkan t dari S, yang digunakan sebagai bagian dari *keystream*. Dapat dilihat pada persamaan (4).

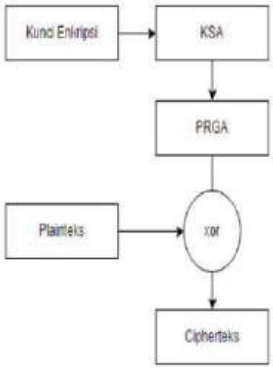
$$i = (i + 1) \bmod 256$$
$$j = (j + S[i]) \bmod 256$$
$$\text{Tukar } S[i] \text{ dengan } S[j]$$
$$t = (S[i] + S[j]) \bmod 256$$
$$K=S[t]; \quad (4)$$

e. *Keystream K* ini kemudian digunakan untuk proses enkripsi dengan melakukan operasi *XOR* antara *keystream* dengan *plaintext* untuk menghasilkan *ciphertext*. Proses dekripsi serupa dengan menggunakan *XOR* antara *ciphertext* dan *keystream* yang sama untuk mendapatkan kembali *plaintext.*

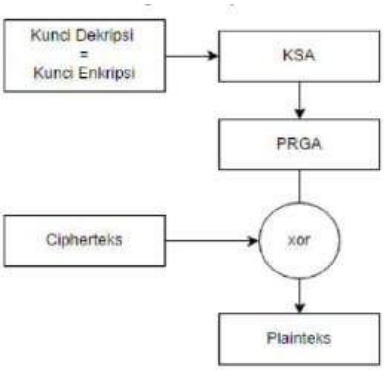
2.3.4 Algoritma *Enkripsi RC4*

Proses dekripsi *RC4* sangat mirip dengan proses enkripsinya, tetapi terdapat perbedaan penting dalam bagaimana *stream* dihasilkan. Untuk mengembalikan *plaintext* dari *ciphertext*, operasi *XOR* dilakukan menggunakan byte *pseudorandom* yang sama yang digunakan dalam proses enkripsi. Tahapan pengaturan kunci pada dekripsi adalah sama persis seperti saat enkripsi, dimulai dengan inisialisasi S-Box, memasukkan kunci ke dalam array *byte* yang spesifik, dan kemudian melakukan inisialisasi ulang *S-Box* berdasarkan array *byte* kunci tersebut[8],[9].

Selama proses dekripsi, *keystream* yang dihasilkan sama persis dengan yang digunakan dalam proses enkripsi, sehingga memastikan bahwa operasi *XOR* dapat mengembalikan *plaintext* dengan tepat. Oleh karena itu, perbedaan antara enkripsi dan dekripsi hanya terletak pada cara *stream* dihasilkan dan digunakan. Dalam dekripsi, kombinasi antara *ciphertext* dan *keystream* melalui operasi *XOR* adalah langkah kritis yang memungkinkan pemulihan *plaintext* asli. Ini menunjukkan bahwa kekuatan enkripsi *RC4* terletak pada kemampuannya untuk menghasilkan *stream* yang seragam dan aman, baik untuk proses enkripsi maupun dekripsi, memastikan integritas dan kerahasiaan data yang dipertukarkan. Pada Gambar 3 dan Gambar 4 berikut, merupakan Ilustrasi dari konsep dekripsi *RC4*[8],[9].



Gambar 3. Konsep Enkripsi *RC4*

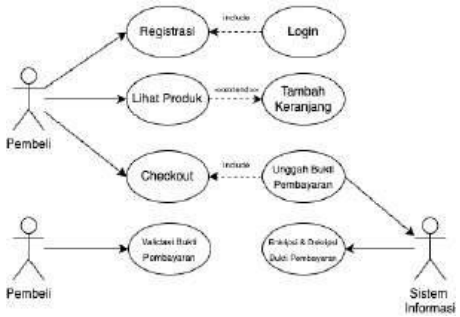


Gambar 4. Konsep Dekripsi *RC4*

2.4 Rancangan Pengujian

2.4.1 *Use case Diagram*

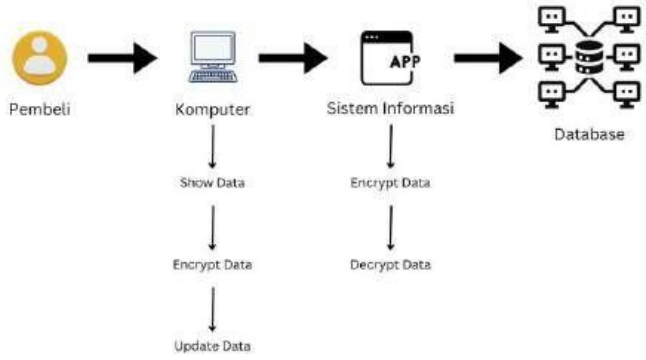
Use case diagram adalah representasi visual dari sebuah sistem yang dilihat dari perspektif pengguna (*user*). Dalam *use case diagram*, pengguna sistem disebut sebagai aktor. Pada Gambar 5 berikut, adalah ilustrasi dari *Use case Diagram* tersebut:



Gambar 5. Use case Diagram

2.4.2 Arsitektur Sistem

Pada Gambar 6 berikut ini, merupakan arsitektur sistem aplikasi gambaran umum untuk dapat memahami konsep aplikasi yang dibangun[10].



Gambar 6. Arsitektur Sistem

3. HASIL DAN PEMBAHASAN

3.1 Lingkungan Percobaan

Sebelum implementasi dan pengujian sistem dapat dilakukan, penting untuk melakukan persiapan komprehensif untuk memastikan bahwa sistem beroperasi sesuai dengan ekspektasi dan rencana yang telah ditetapkan. Tahap persiapan untuk implementasi ini melibatkan langkah-langkah kritis yang harus diikuti :

3.1.1 Spesifikasi Perangkat Keras

Saat pengembangan aplikasi ini, localhost digunakan sebagai platform untuk mendemonstrasikan fungsi aplikasi. Berikut disajikan spesifikasi perangkat keras yang diperlukan untuk menjalankan aplikasi ini dengan optimal.

a. *Processor Intel Core i5* atau *AMD Ryzen 5*
b. *RAM* 8 GB
c. 256 GB *SSD*

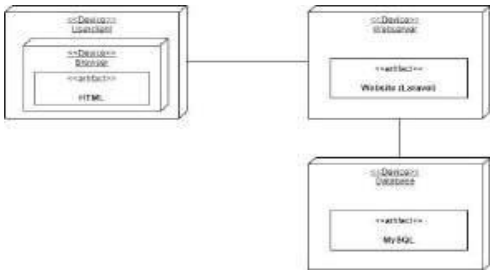
3.1.2 Spesifikasi Perangkat Lunak

Penelitian ini mengandalkan penggunaan berbagai aplikasi perangkat lunak, yang diuraikan sebagai berikut:

a. *Windows 10*
b. *Visual Studio Code*, digunakan sebagai *code editor*
c. *Google Chrome*

3.1.3 *Deployment Diagram*

Dalam *deployment diagram*, *diagram* yang disajikan mengilustrasikan konfigurasi infrastruktur teknis untuk aplikasi yang dikembangkan dengan menggunakan *Laravel* sebagai *framework* pada sisi *server*. *Diagram* ini terdiri dari tiga komponen utama: *User Client***,** *Web Server, dan Database.* Komponen *User Client* menunjukkan perangkat pengguna yang mengakses sistem melalui *browser*, menggunakan *HTML* untuk struktur tampilan pengguna. *Web Server* menggambarkan *server* yang menjalankan aplikasi *web*, dengan *Laravel* sebagai inti dari pengolahan logika bisnis dan data. Terakhir, komponen *Database* menggambarkan penggunaan MySQL sebagai sistem manajemen database untuk penyimpanan data yang efektif dan efisien. *Deployment diagram* ini secara efektif memvisualisasikan bagaimana aplikasi dipetakan pada infrastruktur yang mendukungnya, menyoroti interaksi antara pengguna, *server* dan *database.* Pada Gambar 7 berikut ini, adalah ilustrasi dari *deployment diagram.*



Gambar 7. Deployment Diagram

3.2 Implementasi Metode

Proses implementasi kriptografi pada website *e-commerce* yang menggunakan metode *Caesar Cipher* dan *RC4* dimulai dengan penerapan *Caesar Cipher* untuk mengenkripsi data yang kurang sensitif dengan teknik pergeseran huruf. Selanjutnya, *RC4* diimplementasikan sebagai lapisan kedua untuk mengamankan data yang sangat sensitif, termasuk detail transaksi dan informasi pribadi pengguna. Teknik ini memastikan bahwa data dienkripsi saat meninggalkan perangkat pengguna dan hanya bisa didekripsi di *server* tujuan, memperkuat keamanan data selama dikirimkan melalui jaringan. Penyesuaian dilakukan pada bagian frontend yang mengenkripsi data sebelum pengiriman serta pada *backend* yang bertanggung jawab atas dekripsi data yang diterima. Setelah implementasi, dilakukan pengujian untuk memastikan bahwa fungsi enkripsi dan dekripsi berjalan dengan baik dan tidak mengganggu operasional situs, menunjukkan pentingnya melindungi data pengguna dari akses tidak sah dan ancaman siber lainnya.

3.2.1 Tahap Pengumpulan Data

Dalam tahap pengumpulan data, prosesnya dilakukan dengan menghimpun bukti transfer yang diunggah oleh pengguna di dalam platform e-commerce. Pengumpulan ini

melibatkan menerima berbagai jenis file yang di- *upload* oleh pengguna sebagai bagian dari proses transaksi. File yang dapat diterima dan diolah oleh sistem termasuk *format* gambar populer seperti *JPEG, PNG, JPG, GIF, SVG, DOCX, XLSX* dan *PDF*. Ini memungkinkan sistem untuk mengakomodasi berbagai bentuk bukti pembayaran, memastikan bahwa data transaksi dikumpulkan secara efektif dan aman. Setiap *file* yang di-*upload* kemudian diverifikasi dan dianalisis untuk memastikan keaslian dan keakuratan data transaksi yang disajikan, memfasilitasi analisis lanjutan atau langkah keamanan yang diperlukan untuk proses validasi transaksi.

3.2.2 Tahap Enkripsi

Proses enkripsi dimulai saat pengguna mengunggah bukti bayar ke sistem melalui *platform e-commerce*. Setelah pengguna menekan tombol *submit*, *file* tersebut menjalani enkripsi dua lapis untuk meningkatkan keamanan data. Langkah pertama dalam proses enkripsi ini menggunakan algoritma *RC4*, yang terkenal dengan kemampuannya dalam menghasilkan *stream cipher* yang kuat dan sulit didekripsi tanpa kunci yang tepat. Setelah enkripsi *RC4*, *file* tersebut kemudian dienkripsi lagi menggunakan metode *Caesar Cipher*, yang merupakan teknik substitusi sederhana namun efektif. Hasil dari enkripsi ganda ini adalah file yang sangat aman, yang kemudian disimpan bersama dengan metadata file dalam sistem. Metadata ini mencakup *detail* seperti waktu enkripsi dan *size* dari *file* tersebut, yang semuanya dicatat dalam *log*.

3.2.3 Tahap Dekripsi

Untuk mengakses *file* yang telah dienkripsi, pengguna perlu melalui proses dekripsi yang membalik urutan enkripsi. Pertama, *file* tersebut didekripsi menggunakan *Caesar Cipher* untuk mengembalikan perubahan yang dibuat oleh metode substitusi tersebut. Setelah itu, *file* yang telah di-*decipher* dijalankan melalui proses dekripsi *RC4*, yang mengembalikan *file* ke bentuk aslinya sebelum enkripsi pertama dilakukan. Seperti dalam proses enkripsi, *metadata* dari proses dekripsi termasuk waktu dekripsi dan *size* dari *file* juga dicatat dalam *log*

3.3 Pengujian Program

Dalam proses pengujian program enkripsi dan dekripsi untuk bukti transaksi e-commerce, berbagai sampel bukti transaksi dalam *format JPEG, JPG, SVG, GIF, PNG, DOCX, XLSX, dan PDF* akan diuji untuk memverifikasi beberapa aspek kritis. Pengujian ini mencakup verifikasi bahwa sistem secara akurat mengenali dan memproses berbagai *format file*, serta memastikan integritas metadata yang menyertai *file* tersebut setelah dienkripsi dan didekripsi, memverifikasi tidak ada perubahan yang tidak diinginkan. Selanjutnya, pengujian fokus pada aksesibilitas data pasca-dekripsi untuk memastikan bahwa data dapat diakses dan dibaca dengan benar. Proses enkripsi dan dekripsi itu sendiri juga diperiksa untuk konsistensi dan keandalannya, sementara efisiensi waktu proses diukur untuk menilai performa sistem di bawah beban kerja. Keseluruhan pengujian ini bertujuan untuk menjamin bahwa sistem menawarkan keamanan yang *robust* tanpa mengorbankan kecepatan atau efisiensi, mempersiapkan sistem untuk implementasi yang sukses dan operasi sehari-hari yang aman.

Dalam Pengujian yang telah dilakukan pada data diatas didapatkan hasil pengujian enkripsi dengan rata-rata waktu yang dibutuhkan adalah 0,069003821 detik. Pada Tabel 1 berikut ini, adalah hasil pengujian enkripsi :

Table 1. Tabel Pengujian Enkripsi

No	Dokumen	Waktu (Detik)
1	File 1 (*jpg*)	0,024679184
2	File 2 (*jpeg*)	0,021522999
3	File 3 (*png*)	0,023796082
4	File 4 (*svg*)	0,033878088
5	File 5 (*gif*)	0,022947073
6	File 6 (*docx*)	0,107964993

7	File 7 (*xlsx*)	0,13888216
8	File 8 (*pdf*)	0,178359985
Rata-Rata (Detik)		0,069003821

Lalu dalam pengujian dekripsi sesuai dengan data yang telah di uji di dapatkan rata-rata waktu yang dibutuhkan untuk dekripsi *file* tersebut adalah 0,085177243. Pada Tabel 2 berikut ini, adalah hasil pengujian dekripsi :

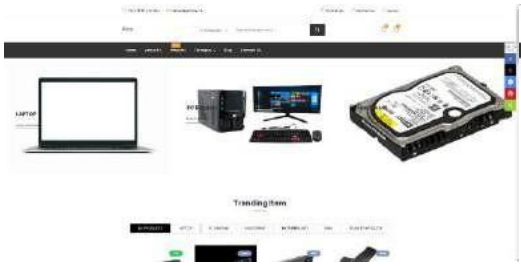
Tabel 2. Tabel Pengujian Dekripsi

No	Dokumen	Waktu (Detik)
1	File 1 (*jpg*)	0,037993908
2	File 2 (*jpeg*)	0,034663916
3	File 3 (*png*)	0,04160285
4	File 4 (*svg*)	0,046261072
5	File 5 (*gif*)	0,046261072
6	File 6 (*docx*)	0,168469906
7	File 7 (*xlsx*)	0,162433147
8	File 8 (*pdf*)	0,143732071
Rata-Rata (Detik)		0,085177243

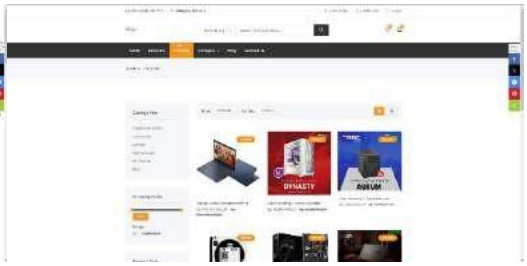
Dari pengujian diatas dapat disimpulkan bahwa penggunaan *Caesar Cipher* dan juga *RC4* sebagai enkripsi data baik dan efisien, dan perlu diperhatikan juga bahwa lama atau cepatnya proses enkripsi data dipengaruhi oleh besarnya *size* atau ukuran dari *file* yang di enkripsi.

3.4 Tampilan Layar Aplikasi

Gambar 8 merupakan Tampilan Layar Menu Utama yang menampilkan *banner* untuk *item trend* seperti *laptop, PC* custom*,* dan *hard drive*, lengkap dengan tombol *"Discover Now"* untuk eksplorasi lebih lanjut. Gambar 9 merupakan Tampilan Layar Produk yang menampilkan berbagai kategori barang elektronik dengan tawaran harga menarik. Antarmuka ini memudahkan pengguna untuk menjelajahi dan membandingkan produk seperti *laptop*, komponen *PC*, dan perangkat penyimpanan. Setiap item dilengkapi dengan *detail* harga dan diskon yang jelas, serta pilihan untuk mengurutkan dan mem-*filter* produk berdasarkan harga.



Gambar 8. Tampilan Layar Menu Utama



Gambar 9. Tampilan Layar Produk

Gambar 10 merupakan Tampilan Layar *Detail* Produk. Halaman detail produk ini menampilkan *laptop Lenovo* Ideapad *Slim 3*, tersedia dengan harga promosi. Pengunjung dapat menyesuaikan jumlah pembelian, memeriksa ketersediaan stok, dan membaca deskripsi serta ulasan produk sebelum menambahkan *item* ke keranjang belanja. Gambar 11 merupakan Tampilan Layar Kategori. Halaman kategori pada *website e-commerce* ini memperlihatkan beragam produk elektronik seperti *laptop, hard drive,* dan *motherboard.* Pengguna dapat melihat dan membandingkan produk secara langsung dengan informasi harga dan *detail* singkat disertakan di bawah setiap gambar. Fitur sortir dan *filter* memudahkan pencarian produk berdasarkan harga atau spesifikasi, sementara opsi *"Buy Now!"* memfasilitasi proses pembelian yang cepat dan mudah.

Gambar 10. Tampilan Layar Detail Produk

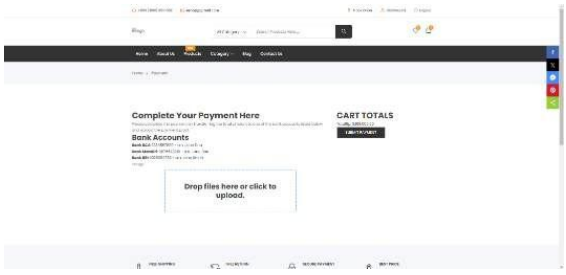
Gambar 11. Tampilan Layar Kategori

Gambar 12 merupakan Tampilan Layar *Cart* yang menampilkan produk yang dipilih oleh pengguna, dengan informasi men-*detail* seperti nama, harga per *unit*, dan jumlah. Pengguna dapat meng*update* jumlah produk atau melanjutkan berbelanja untuk menambah *item* lain. Proses *checkout* dapat dilakukan dengan mudah melalui tombol *"Checkout"*. Gambar 13 merupakan Tampilan Layar *Checkout* yang memudahkan pengguna untuk melengkapi pembelian dengan mengisi data pribadi, seperti nama, *e-mail*, dan alamat pengiriman. Pengguna dapat memilih metode pembayaran yang diinginkan, termasuk opsi bayar saat pengiriman atau *transfer*. Ringkasan keranjang belanja menunjukkan total biaya, termasuk pengiriman gratis, dengan tombol *'PROCEED TO CHECKOUT'*.

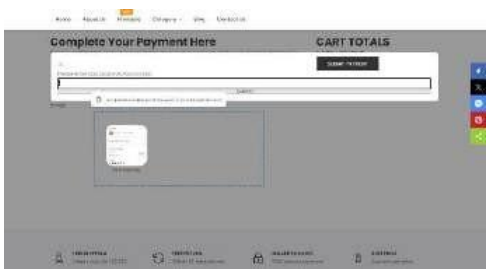
Gambar 12. Tampilan Layar Cart

Gambar 13. Tampilan Layar Checkout

Gambar 14 merupakan Tampilan Layar *Upload Payment* yang memungkinkan pengguna untuk menyelesaikan transaksi mereka dengan mengisi informasi pembayaran dan mengunggah bukti pembayaran. Pengguna dapat memilih dari beberapa rekening *bank* yang tersedia untuk *transfer* dana dan mengunggah bukti pembayaran langsung di halaman ini melalui fitur *drag-and-drop*. Gambar 15 merupakan Tampilan Layar *Key* Enkripsi yang bertujuan untuk memasukkan *Key* untuk Enkripsi bukti bayar.

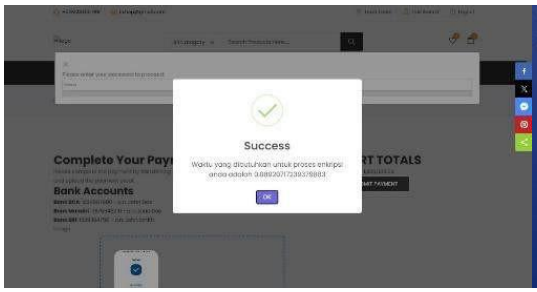


Gambar 14. Tampilan Layar Upload Payment

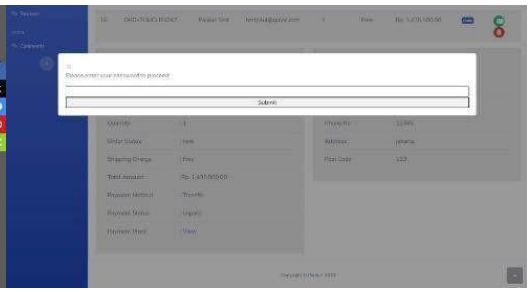


Gambar 15. Tampilan Layar Key Enkripsi

Gambar 16 merupakan Tampilan Layar *Alert* Enkripsi yang berfungsi memunculkan tampilan *alert* untuk memperlihatkan waktu yang dibutuhkan dalam meng-enkripsi file bukti bayar yang telah di *upload user*. Gambar 17 merupakan Tampilan Layar *Key* Dekripsi yang bertujuan untuk memasukkan *key* yang sama yang tadi digunakan untuk meng-enkripsi bukti bayar. Hal bertujuan untuk menjaga kerahasiaan data yang di enkripsi.



Gambar 16. Tampilan Layar Alert Enkripsi



Gambar 17. Tampilan Layar Key Dekripsi

4. KESIMPULAN

Setelah melaksanakan penelitian mengenai implementasi kriptografi dengan enkripsi *end-to-end* untuk melindungi informasi transaksi pada *website e-commerce* menggunakan metode *Caesar Cipher* dan *RC4*, kesimpulan yang dapat peneliti sampaikan berdasarkan rumusan masalah dan tujuan penelitian bahwa metode *Caesar Cipher* dan *RC4* terbukti efektif dan efisien dalam mengenkripsi dan melindungi informasi transaksi. Penggunaan kedua metode enkripsi ini menyediakan tingkat keamanan yang baik dalam lingkungan *e-commerce*. Dan selama fase pengujian, metode ini diuji pada data bukti bayar dalam lima *format file* berbeda. Pengujian menunjukkan bahwa waktu enkripsi rata-rata adalah 0,069003821 detik, sedangkan waktu dekripsi rata-rata adalah 0,085177243 detik. Hasil ini menunjukkan bahwa proses enkripsi dan dekripsi sangat cepat, namun perlu diperhatikan bahwa kecepatan proses ini dapat dipengaruhi oleh ukuran *file* yang dienkripsi.

Berdasarkan hasil penelitian pada implementasi kriptografi dengan metode Caesar Cipher dan RC4 untuk melindungi informasi transaksi di sebuah platform e-commerce, beberapa saran untuk pengembangan lebih lanjut adalah sebagai menggunakan variasi data yang lebih luas, termasuk berbagai format file dan ukuran, untuk menguji keandalan dan efisiensi metode enkripsi dalam skenario penggunaan nyata yang beragam, mengintegrasikan sistem enkripsi ini ke dalam berbagai aplikasi atau sistem lain untuk mengevaluasi adaptabilitas dan efektivitasnya dalam konteks yang berbeda, memastikan bahwa enkripsi dapat diaplikasikan secara luas dengan efisiensi yang tinggi, dan memperhatikan faktor-faktor yang mempengaruhi kecepatan enkripsi dan dekripsi, seperti ukuran file, dan mengoptimalkan algoritma untuk menangani file berukuran besar lebih cepat, sehingga meningkatkan kinerja keseluruhan sistem dalam mengelola transaksi yang aman.

5. DAFTAR PUSTAKA

[1] Rahmadhiyanti, S. “Implementasi Kriptografi RSA Untuk Peningkatan Keamanan Database E-Commerce”. *Jurnal Pelita Informatika*, 2019.

[2] Nur Faqih, F., Tahir, M., Ashfarina, Z., Irsyad Faa, R., Alfarisi, S., & Erfani, F. “Efektivitas Peningkatan Keamanan *Login* Pada *Website* Menggunakan Enkripsi *Caesar Chipper”*. *ADIJAYA Jurnal Multidisiplin*. https://e-journal.naureendigition.com/index.php/mj, 2023.

[3] Tamarahadi, R & Ragam, R. “Implementasi Kriptografi Keamanan File Menggunakan Algoritme *Advanced Encryption Standard 128* Berbasis Web”. 3rd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI), 2023.

[4] Andrea, K., Wardana, A., Wanandi, B. S., & Ikhwan, A. “Penerapan Kriptografi *Caesar Cipher* Pada Fitur Aplikasi Chatting Whatsapp.” *Jurnal Hasi Penelitian dan Pengkajian Ilmiah Eksakta*, 2(1), 6. https://doi.org/10.47233/jppie.v2i1.660, 2023.

[5] Dwi Putri, Y., Lutfi, S., Jati Metro, J., & Ternate Selatan, K. “Penerapan Kriptografi *Caesar Cipher* Pada Fitur *Chatting*

[6] Sistem Informasi *Freelance”*. *Jurnal Informatika Dan Komputer*, 2(2), 2355–7699. https://doi.org/10.33387/jiko, 2019.

[7] Hidayat, M & Tahir, M. "Penerapan Kriptografi *Caesar Chiper* Dalam Pengamanan Data". *JURNAL JUKIM* Vol 2 No. 3 Mei 2023.
[8] Arya, D., Virgian, D., & Sakti, S. Y. "Implementasi Algoritma Kriptografi *Rivest Code 4 (RC4)* Berbasis Web Pada PT. Putri Maharani Medikal". In Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia, 2022.
[9] Darmawan, S., & Imelda. "Pengamanan Dokumen Menggunakan Kriptografi *RC4* dan Steganografi EOF dengan Media
[10] *Video MP4* pada CV. Synergy Selaras". *Jurnal Teknologi Elektro*, 8(2), 117, 2017.
[11] Muharyanto, A. S., & Fatimah, T. "Keamanan *Database* Dengan Metode *Rivest Code 4 (RC4)* dan *Caesar Cipher*
[12] Berbasis Desktop". *SKANIKA: Sistem Komputer Dan Teknik Informatika*, 1(2), 508–513, 2018.
[13] Oloan Simamora, D. P. "Implementasi Algoritma RC4 dan Playfair Cipher Untuk Mengamankan Data Teks". *Jurnal Pelita Informatika*, 2017.