SENAFTI
SEMINAR NASIONAL MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI

6th Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
20 September 2025 – Jakarta, Indonesia
Volume 4, Nomor 2, September 2025 - ISSN 2962-8628 (*online*)

IMPLEMENTASI KRIPTOGRAFI ALGORITMA *VIGENERE CIPHER* DAN RC4 MODIFIKASI UNTUK MENGAMANKAN DATA

Galih Sadewo[1*], Dolly Virgian Shaka Yudha Sakti[2]

[1,2]Teknik Informatika, Teknologi Informasi, Universitas Budi Luhur, Jakarta Selatan, Indonesia

Email: [1*]galihsadewo02@gmail.com, [2]dolly.virgianshaka@budiluhur.ac.id

Abstrak- Meningkatnya penggunaan toko *online* di era *digital* menimbulkan ancaman serius terhadap kebocoran data pelanggan, terutama pada *file* yang berisi informasi sensitif seperti identitas dan detail transaksi. Untuk menjawab permasalahan tersebut, penelitian ini mengembangkan aplikasi enkripsi dan dekripsi *file* berbasis *web* dengan menggabungkan algoritma *Vigenere Cipher* dan RC4 yang telah dimodifikasi. Modifikasi RC4 dilakukan dengan mengulang proses *Key Scheduling Algorithm* (KSA) sebanyak tiga kali serta membuang 768 *byte* awal dari *keystream* guna mengurangi bias statistik. Aplikasi diuji pada *file Excel* (.xlsx) dengan ukuran bervariasi, mulai dari 11 KB hingga 9 MB. Hasil implementasi menunjukkan bahwa *file* berhasil dienkripsi menjadi *ciphertext* yang hanya dapat dikembalikan dengan kunci yang benar. Uji coba fungsional memastikan semua fitur berjalan sesuai rancangan, uji keamanan membuktikan algoritma sangat sensitif terhadap perubahan kunci (satu karakter berbeda menyebabkan kegagalan dekripsi), sedangkan uji performa memperlihatkan waktu enkripsi dan dekripsi relatif singkat, yakni 0,07 detik untuk *file* 11 KB dan 2,4 detik untuk *file* 9 MB.Dengan demikian, penelitian ini membuktikan bahwa metode enkripsi dua lapis yang ringan namun efektif dapat meningkatkan keamanan data pelanggan pada aplikasi toko *online*. Secara ilmiah, hasil ini dapat menjadi alternatif solusi kriptografi sederhana namun efisien yang relevan untuk pengamanan data pada platform *e-commerce* berskala kecil hingga menengah.

Kata Kunci**:** Kriptografi, Vigenere Cipher, RC4 Modifikasi, Enkripsi File

IMPLEMENTATION OF VIGENERE CIPHER AND MODIFIED RC4 ALGORITHM CRYPTOGRAPHY TO SECURE DATA

Abstract*— The increasing use of online stores in the digital era poses a serious threat of customer data leakage, particularly in file s containing sensitive information such as personal identity and transaction details. To address this issue, this study developed a web-based file encryption and decryption application by combining the Vigenère Cipher algorithm and a modified RC4. The RC4 modification was carried out by repeating the Key Scheduling Algorithm (KSA) process three times and discarding the first 768 byte s of the keystream to reduce statistical bias. The application was tested on Excel (.xlsx) file s of varying sizes, ranging from 11 KB to 9 MB. The implementation results showed that the file s were successfully encrypted into ciphertext that could only be decrypted using the correct key. Functional testing confirmed that all features operated as designed, security testing demonstrated that the algorithm is highly sensitive to key changes (a single character difference resulted in decryption failure), while performance testing showed relatively short encryption–decryption times, namely 0.07 seconds for an 11 KB file and 2.4 seconds for a 9 MB file . Thus, this research proves that a lightweight yet effective two-layer encryption method can enhance customer data security in online store applications. Scientifically, the results can serve as an alternative cryptographic solution that is both simple and efficient, particularly relevant for securing data in small- to medium-scale e-commerce platforms.*

Keywords: Cryptography, Vigenère Cipher, Modified RC4, File Encryption

---

1. PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan signifikan pada berbagai sektor [1], termasuk perdagangan elektronik (*e-commerce*). Layanan belanja *online* semakin diminati masyarakat karena kemudahan akses, kecepatan transaksi, dan ragam pilihan produk yang tersedia. Namun, di balik kenyamanan tersebut, terdapat ancaman serius terhadap keamanan data pelanggan. Informasi sensitif seperti nomor telepon, alamat rumah, dan detail transaksi sering kali dibagikan saat proses pembelian. Apabila data ini tidak dikelola dengan benar, risiko kebocoran informasi menjadi sangat tinggi dan dapat memicu tindak kriminal seperti penipuan maupun pencurian identitas [2].

Penelitian terdahulu di Indonesia menunjukkan bahwa pengamanan data pelanggan pada toko *online* umumnya berfokus pada proteksi sistem jaringan, penggunaan *firewall*, serta metode autentikasi dua faktor [3]. Meskipun pendekatan tersebut efektif untuk mencegah akses ilegal pada akun pengguna, hanya sedikit penelitian yang secara spesifik membahas perlindungan *file* statis yang menyimpan data pelanggan, khususnya dalam konteks transaksi *e-commerce* [4]. Berbeda dengan penelitian sebelumnya yang menitikberatkan pada proteksi

jaringan, penelitian ini berfokus pada perlindungan *file* statis melalui penerapan enkripsi dua lapis. Enkripsi berlapis di pilih karena Algoritma tunggal sering kali tidak cukup untuk mengatasi risiko ini, terutama dalam lingkungan yang rawan serangan siber dan akses tidak sah [5].

Kasus pada salah satu toko *online* yang diteliti memperlihatkan lemahnya sistem pengamanan data. Berdasarkan riset awal, ditemukan bahwa perusahaan ini tidak memiliki mekanisme keamanan untuk *file* yang berisi informasi pribadi pelanggan. Kondisi tersebut menunjukkan adanya celah besar dalam pengelolaan data, sehingga sangat rentan terhadap ancaman kebocoran yang berpotensi menurunkan reputasi bisnis dan kepercayaan pelanggan.

Salah satu solusi yang dapat diterapkan adalah penggunaan teknik kriptografi. Kriptografi berperan penting dalam menjaga kerahasiaan dan integritas data dengan mengubah *plaintext* menjadi *ciphertext* yang hanya dapat dikembalikan ke bentuk semula menggunakan kunci yang sesuai. Algoritma klasik seperti *Vigenere Cipher* dikenal sederhana dan ringan, sedangkan algoritma stream cipher RC4 memiliki keunggulan dari sisi kecepatan [6]. Meski demikian, keduanya memiliki kelemahan: *Vigenere Cipher* rentan terhadap analisis pola, sedangkan RC4 standar memiliki bias statistik pada keluaran awal *keystream* [7].

Untuk mengatasi kelemahan tersebut, penelitian ini mengusulkan metode enkripsi dua lapis dengan mengombinasikan *Vigenere Cipher* berbasis *byte* dan RC4 yang telah dimodifikasi. Modifikasi RC4 dilakukan melalui pengulangan proses *Key Scheduling Algorithm* (KSA) sebanyak tiga kali dan pembuangan 768 *byte* awal pada *Pseudo-Random Generation Algorithm* (PRGA). Pendekatan ini diharapkan dapat menghasilkan *keystream* yang lebih acak sehingga meningkatkan keamanan tanpa mengorbankan performa sistem.

Dengan demikian, tujuan penelitian ini adalah mengembangkan aplikasi enkripsi dan dekripsi *file* berbasis *web* yang mampu mengamankan *file* berformat *Excel* berisi data pelanggan pada toko *online* yang diteliti, dengan memanfaatkan kombinasi *Vigenere Cipher* dan RC4 Modifikasi sebagai inti algoritma.

2. METODE PENELITIAN

2.1. Data Penelitian

Data yang digunakan dalam penelitian ini berupa *file digital* berformat Microsoft *Excel* (.xlsx) yang berisi informasi pelanggan seperti nama lengkap, alamat, nomor telepon, nomor pesanan, dan nomor resi pengiriman. Karena mempertimbangkan aspek kerahasiaan, data asli tidak digunakan. Sebagai gantinya, dibuat 14 *file* simulasi dengan struktur dan format yang menyerupai data asli, namun berisi data fiktif Ukuran *file* bervariasi mulai dari puluhan kilo*byte* hingga beberapa mega*byte.* Variasi ini bertujuan untuk menguji performa sistem pada kondisi yang mendekati penggunaan nyata.

2.2. Algoritma yang Digunakan

Penelitian ini memanfaatkan kombinasi *Vigenere Cipher* dan RC4 Modifikasi, yang dipadukan dalam proses enkripsi berlapis. Pemilihan dua algoritma ini didasarkan pada pertimbangan bahwa *Vigenere Cipher* memiliki sifat polyalphabetic substitution cipher yang mampu memecah pola data awal, sementara RC4 sebagai stream cipher unggul dari segi kecepatan pemrosesan.

2.2.1. *Vigenere Cipher*

Vigenere Cipher adalah algoritma kriptografi klasik yang termasuk dalam kategori *polyalphabetic substitution cipher*, di mana setiap karakter pada *plaintext* dienkripsi menggunakan kunci yang berulang [8]. Versi klasik algoritma ini hanya dapat mengenkripsi huruf alfabet (A–Z), sehingga kurang cocok untuk *file digital* yang mengandung *byte* di luar rentang alfabet [9].

Dalam penelitian ini, algoritma *Vigenere Cipher* dirubah agar dapat memproses data berbasis *byte* , sehingga bisa mengenkripsi *file* biner seperti *Excel* (.xlsx). Prosesnya sebagai berikut:

a. Enkripsi

Setiap *byte plaintext* (Mi) dijumlahkan dengan *byte* kunci (Ki) sesuai urutan, lalu dilakukan operasi *modulo* 256:

$$Ci = (Mi + Ki) \bmod 256 \quad (1)$$

Proses diulang secara sirkular hingga seluruh *byte file* terenkripsi.

b. Dekripsi

Setiap *byte ciphertext* (Ci) dikurangi dengan *byte* kunci (Ki) lalu ditambah 256 (untuk menghindari bilangan negatif), kemudian dilakukan *modulo* 256:

$$Mi = (Ci - Ki + 256) \bmod 256 \quad (2)$$

c. RC4 Modifikasi

RC4 adalah algoritma *stream cipher* simetris [10] yang bekerja dengan dua tahap utama:

1. *Key Scheduling Algorithm* (KSA)
   Menginisialisasi dan mengacak array status S[0..255] berdasarkan kunci. Dengan rumus :
   $$j = (j + S[i] + K[i \bmod k]) \bmod 256 \quad (3)$$

2. *Pseudo-Random Generation Algorithm* (PRGA)
   Menghasilkan *keystream* yang akan di-*XOR* dengan *plaintext* atau *ciphertext* .
   Dengan rumus :

$$i=(i+1) \bmod 256 \quad (4)$$

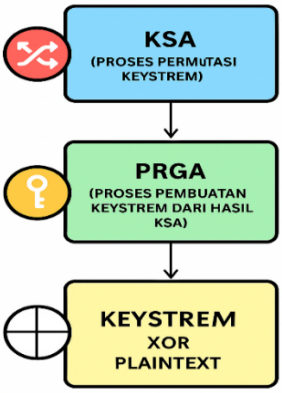
$$j = (j + S[i]) \bmod 256 \quad (5)$$

$$Tukar\ S[i]\ dengan\ S[j] \quad (6)$$

$$Z = S[(S[i] + S[j]) \bmod 256] \quad (7)$$

Meskipun cepat, RC4 standar memiliki kelemahan pada keluaran awal *keystream* , di mana terdapat bias statistik yang membuat sebagian *byte* awal lebih mudah diprediksi [7].

Secara garis besar alur algoritma RC4 standar berupa proses permutasi *keystrem* yang biasa di sebut KSA kemudian di lanjutkan dengan pembuatan *keystrem* yang akan digunakan untuk proses XOR dengan *plaintext* di proses selanjutnya. Proses ini dapat di lihat pada gambar 1. Alur Algoritma RC4 Standar



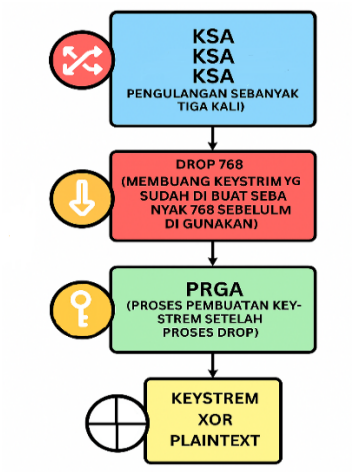
Gamabr 1. Alur Algoritma RC4 Standar

Pada penelitian ini menerapkan dua bentuk modifikasi untuk mengurangi kelemahan tersebut:

a. Multiple KSA
   Proses KSA dijalankan tiga kali dengan kunci yang sama. Tujuannya adalah untuk meningkatkan keacakan distribusi nilai pada array status, sehingga mengurangi kemungkinan pola yang dapat dieksploitasi.

b. Drop[N]
   Setelah KSA selesai, PRGA dijalankan tetapi hasil 768 *byte* awal dibuang (*drop 768*). Hal ini dilakukan karena *byte* awal RC4 cenderung memiliki bias dan kurang acak.

Proses modifikasi dilakukan pada KSA dimana dilakukan proses permutasi diulang hingga tiga kali guna menambah keacakan *keystrem*, kemudian juga ditambahkan proses Drop[N] *keystrem* yaitu pembuangan 768 biye awal *keystrem* yang telah dibuat dengan proses PRGA, setelah 768 *byte* *keystrem* dibuang *keystream* yang dihasilkan setelah proses pembuangan baru akan digunakan pada proses XOR proses pembuatan keystem ini disebut dengan proses PRGA, kemudian keystem dilakukan XOR dengan *plaintext* seperti algoritma standar. Secara keseluruhan alur algoritma RC4 modifikasi dapat di lihat pada gambar 2. Alur Algoritma RC4 Modifikasi



Gambar 2. Alur Algoritma RC4 Modifikasi

2.3. Tahapan Penelitian

Tahapan penelitian mengikuti alur logis mulai dari analisis kebutuhan hingga evaluasi hasil. Tahapan tersebut meliputi:

a. Analisis Kebutuhan
Mengidentifikasi masalah pada sistem pengelolaan data pelanggan *By Barra.Outfit* yang belum memiliki proteksi *file* . Kemudian menentukan solusi berupa enkripsi dua lapis dengan *Vigenere Cipher* dan RC4 Modifikasi.

b. Perancangan Sistem
Menyusun rancangan basis data untuk menyimpan data pengguna, riwayat enkripsi, dan riwayat dekripsi. Mendesain antarmuka pengguna untuk *Login*, unggah *file* , proses enkripsi, proses dekripsi, dan penyimpanan hasil.

c. Implementasi Algoritma
Enkripsi:
1. Hash kunci teks → SHA-256.
2. Enkripsi menggunakan Vigenere Cipher.
3. Tambahkan *magic header* untuk verifikasi saat dekripsi.
4. Enkripsi hasil dengan RC4 Modifikasi.
5. Simpan *file* hasil enkripsi dan meta data ke database.

Dekripsi**:**
1. Hash kunci teks → SHA-256.
2. Dekripsi dengan RC4 Modifikasi.
3. Verifikasi *magic header*.
4. Dekripsi dengan Vigenere Cipher.
5. Simpan *file* hasil dekripsi dan meta data.

d. Pengujian Sistem
Pengujian dilakukan dengan tiga jenis pengujian:
1. Pengujian Fungsionalitas
Pengujian fungsional dilakukan dengan metode *black-box testing* untuk memastikan bahwa setiap fitur bekerja sesuai spesifikasi yang ditetapkan. Beberapa skenario uji yang dilakukan antara lain:

a) *Login* pengguna dengan menguji keberhasilan *Login* dengan kombinasi *username* dan *password* yang benar.
b) Validasi *Login* dengan memastikan sistem menolak akses ketika *username* atau *password* salah.
c) Unggah *file* dengan menguji proses input *file Excel* dengan format dan ukuran sesuai ketentuan.
d) Enkripsi *file* dengan memastikan *file* valid dengan kunci benar dapat dienkripsi dan metadata tersimpan di basis data.
e) Dekripsi dengan kunci salah dengan memastikan sistem menolak dekripsi apabila kunci tidak sesuai.
f) Dekripsi dengan kunci benar dengan memastikan *file* terenkripsi dapat kembali ke bentuk semula jika kunci valid.
g) Validasi ukuran *file* dengan memastikan sistem menolak *file* dengan ukuran melebihi batas maksimum (misalnya >10 MB).

2. Pengujian Keamanan

Pengujian keamanan difokuskan pada sensitivitas kunci (*Key Sensitivity Test*). *File* yang telah terenkripsi dengan kunci asli diuji kembali menggunakan kunci yang diubah sedikit, seperti:

a) mengubah huruf kapital menjadi kecil,
b) menambah atau menghapus satu karakter
c) mengganti satu karakter di tengah kunci. Semua variasi menghasilkan kegagalan dekripsi, sehingga membuktikan algoritma memiliki tingkat sensitivitas tinggi terhadap perubahan kunci.

3. Pengujian Performa

Pengujian performa dilakukan dengan mengukur waktu eksekusi proses enkripsi dan dekripsi menggunakan fungsi *timestamp* pada sistem. Hasil pengukuran ditampilkan dalam milidetik (ms) untuk memperoleh nilai yang lebih presisi. *File* uji memiliki ukuran bervariasi, mulai dari kecil (11 KB), sedang (500 KB), hingga besar (9 MB).

3. HASIL DAN PEMBAHASAN

3.1. Hasil Pengujian Fungsionalitas

Pengujian fungsionalitas bertujuan memastikan bahwa seluruh fitur aplikasi berjalan sesuai spesifikasi. Hasil pengujian ditunjukkan pada Tabel 1

Tabel 1. Hasil Pengujian Fungsionalitas

No	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Uji
1	*Login* pengguna	*Username* & *password* benar	Berhasil masuk ke halaman *home*	Sesuai
2	Validasi *Login*	*Username* & *password* salah	Gagal masuk, muncul pesan *error*	Sesuai
3	Enkripsi *file*	*File* valid, kunci benar	*File* terenkripsi, metadata tersimpan	Sesuai
4	Validasi ukuran *file*	*File* > 10 MB	Ditolak dengan pesan peringatan	Sesuai
5	Dekripsi kunci salah	*File* terenkripsi, kunci salah	Gagal dekripsi, muncul pesan *error*	Sesuai
6	Dekripsi kunci benar	*File* terenkripsi, kunci benar	*File* kembali ke bentuk asli	Sesuai

Hasil ini menunjukkan bahwa sistem mampu memvalidasi *Login*, ukuran *file* , serta kebenaran kunci, sehingga meminimalkan kesalahan penggunaan.

3.2. Hasil Pengujian Keamanan (*Key Sensitivity Test*)

Pengujian sensitivitas kunci dilakukan untuk memastikan bahwa perubahan sekecil apapun pada kunci enkripsi akan menyebabkan kegagalan dekripsi total.
Pada skenario ini, *file* yang terenkripsi dengan kunci asli dideskripsi menggunakan kunci yang dimodifikasi, seperti:

a. Mengubah satu huruf
b. Mengubah huruf kapital menjadi huruf kecil
c. Menambah atau menghapus satu karakter.

Berikut adalah tabel yang menyajikn daftar hasil pengujian keamanan :

Tabel.2 Hasil pengujian keamanan

No	Kunci asli	Jenis perubahan	hasil	keterangan
1	*password* 123	Merubah huruf kapital menjadi kecil	Tidak bisa dekripsi	Kunci sensitive

2	*Password* 123+spasi	Menambah spasi di akhir kunci	Tidak bisa dekripsi	Kunci sensitive
3	*Password* 124	Melompati urutan angka	Tidak bisa dekripsi	Kunci sensitive
4	Pasaword123	Ganti satu huruf di tengah	Tidak bisa dekripsi	Kunci sensitive

Hasil pengujian menunjukkan bahwa seluruh percobaan menghasilkan *file* yang tidak terbaca Hal ini membuktikan bahwa sistem memiliki tingkat sensitivitas kunci yang tinggi.

3.3. Hasil Pengujian Performa

Pengujian performa mengukur waktu proses enkripsi dan dekripsi terhadap *file* dengan berbagai ukuran. Hasil uji dapat diringkas pada tabel berikut:

Tabel 3. Hasil Pengujian Performa Algoritma RC4 Modifikasi

Ukuran *File* (.xlsx)	Waktu Enkripsi (detik)	Waktu Dekripsi (detik)
11 KB	0.0746 detik	0.1337 detik
500 KB	0.3184 detik	0.3541 detik
1 MB	0.4759 detik	0.7406 detik
9 MB	2.4537 detik	2.7932 detik

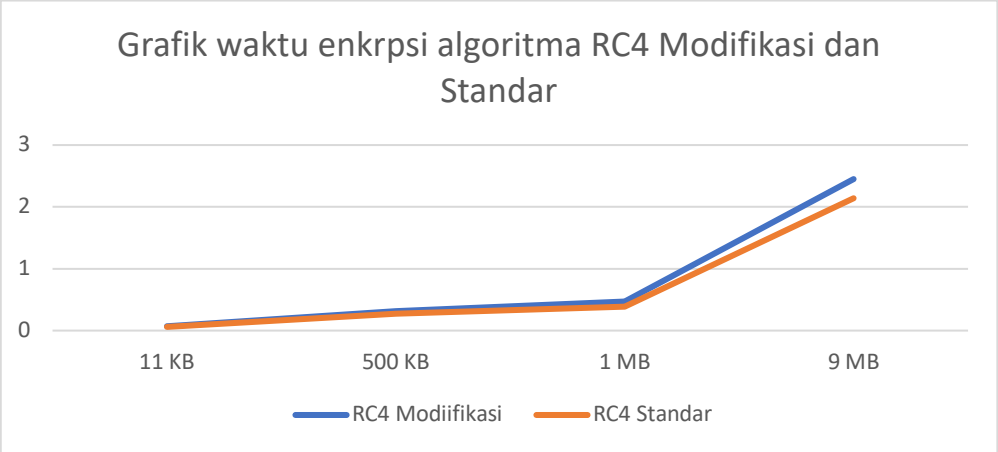
Hasil pengujian menunjukkan bahwa penerapan algoritma RC4 yang telah dimodifikasi dengan mekanisme multiple *Key Scheduling Algorithm* (KSA) dan penerapan drop 768 tidak menimbulkan beban komputasi yang signifikan, meskipun ukuran *file* yang diuji mengalami peningkatan hingga mencapai 9 MB. Dengan demikian, dapat disimpulkan bahwa modifikasi tersebut tidak hanya mampu meningkatkan aspek keamanan, tetapi juga tetap mempertahankan efisiensi proses enkripsi maupun dekripsi.

Selain itu, untuk memperoleh gambaran yang lebih komprehensif mengenai performa sistem, dilakukan pula pengujian perbandingan dengan menggunakan algoritma RC4 standar. Pengujian dilakukan pada *file* dengan ukuran yang sama seperti pada pengujian algoritma RC4 modifikasi, sehingga hasil yang diperoleh dapat dijadikan tolok ukur objektif mengenai perbedaan kinerja di antara kedua algoritma. Hasil perbandingan kinerja tersebut secara rinci ditampilkan pada tabel berikut, yang menggambarkan perbedaan waktu proses enkripsi dan dekripsi antara RC4 standar dan RC4 modifikasi.

Tabel 4. Hasil Pengujian Performa Algoritma RC4 Standar

Ukuran *File* (.xlsx)	Waktu Enkripsi (detik)	Waktu Dekripsi (detik)
11 KB	0.0682 detik	0.1462 detik
500 KB	0.2783 detik	0.3104 detik
1 MB	0.3925 detik	0.5013 detik
9 MB	2.1437 detik	2.5257 detik

Berdasarkan kedua tabel hasil pengujian performa algoritma RC4 modifikasi dan standar, disajikan grafik waktu enkripsi yang menunjukkan adanya peningkatan durasi proses seiring dengan bertambahnya ukuran *file* , mulai dari ukuran 11 KB yang memerlukan waktu kurang dari 1 detik hingga ukuran 9 MB yang membutuhkan waktu lebih dari 2 detik. Secara keseluruhan, hasil tersebut dapat diamati pada grafik berikut:



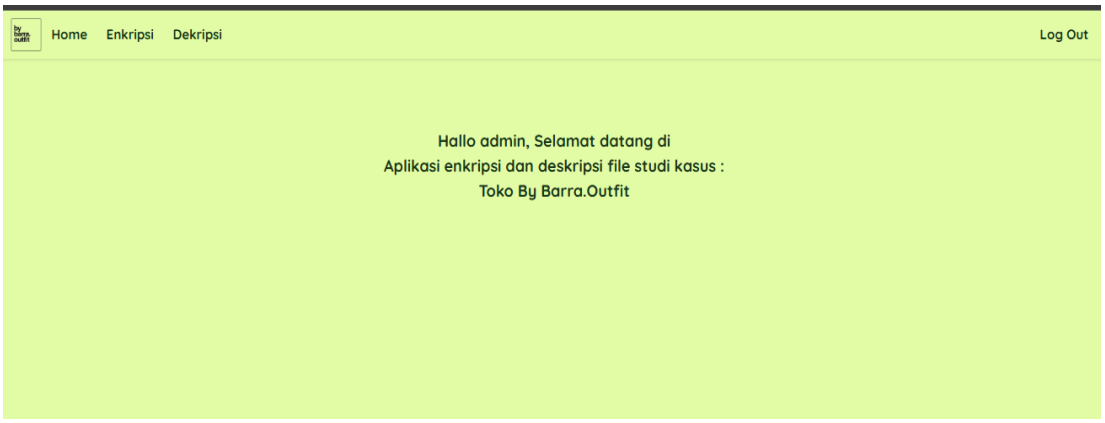
Gambar 5. Grafik waktu enkrpsi algoritma RC4 Modifikasi dan Standar

Dari hasil pengujian performa pada algoritma RC4 Modifikasi dan standar menunjukan bahwa waktu proses Algoritma RC4 modifikasi mengalami peningkatan meskpiun tidak terlalu signifikan. Dapat disimpulkan bahwa penerapan modifikasi pada algortima RC4 masih bisa diimplementasikan karena peningkatan waktu proses tidak terlau signifikan.

3.4. Tampilan Layar

3.4.1. Halaman *Home*

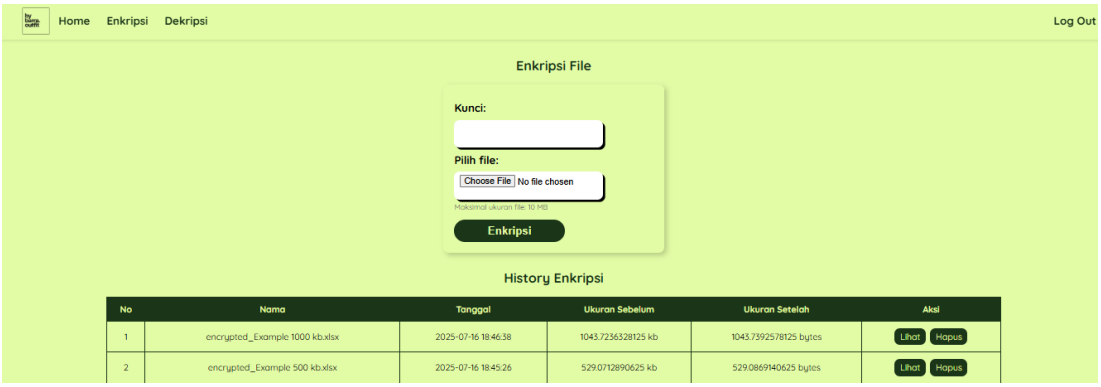
Tampilan halaman beranda pada aplikasi, di bagian atas di sediakan menu navigasi dan ucapan selamat dating pada bagian tengah untuk menandakan pengguna berhasil untuk masuk ke aplikasi



Gambar 6. Halaman *Home*

3.4.2. Halaman Enkripsi

Tampilan Halaman untuk pengguna melakukan proses enkripsi, di sediakan input untuk memasukan *password* dan *file* , serta daftar tabel berisi meta data *file* yang telah terenkripsi sebelumnnya



Gambar 7. Halaman Enkripsi

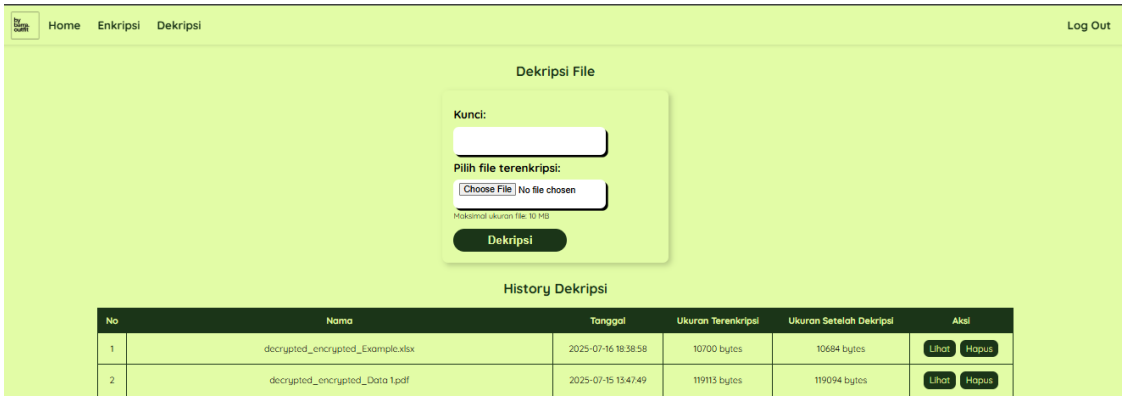
3.4.3. Halaman Hasil Enkripsi

Halaman menunjukan detail informasi dari proses enkripsi yang telah berhasil di lakukan, informasi meliputi nama *file* , tanggal, waktu dan ukuran *file* baik sesudah mapun sebelum

Gambar 8. Halaman Hasil Enkripsi

3.4.4. Halaman Dekripsi

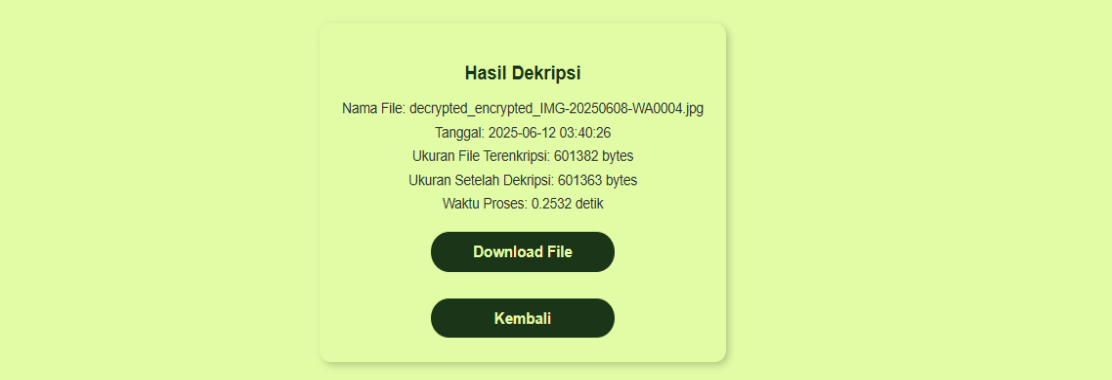
Tampilan Halaman untuk pengguna melakukan proses dekripsi, di sediakan input untuk memasukan *password* dan *file* , serta daftar tabel berisi meta data *file* yang telah terdekripsi sebelumnya



Gambar 9. Halaman Dekripsi

3.4.5. Halaman Hasil Dekripsi

Halaman menunjukan detail informasi dari proses dekripsi yang telah berhasil di lakukan, informasi meliputi nama *file* , tanggal, waktu dan ukuran *file* baik sesudah mapun sebelum.



Gambar 10. Halaman Hasil Dekripsi

3.5 Analisa Hasil Pengujian

Berdasarkan serangkaian percobaan yang telah dilakukan, dapat disimpulkan bahwa aplikasi yang dikembangkan mampu berfungsi dengan baik dan benar sesuai dengan spesifikasi dan fungsi yang dibutuhkan. Seluruh fitur yang diujikan berjalan sesuai harapan tanpa ditemukan kesalahan yang signifikan. Selain itu, dari hasil pengujian keamanan, aplikasi terbukti memiliki ketahanan yang baik terhadap perubahan sensitivitas kunci, sehingga dapat meminimalkan potensi gangguan atau serangan yang memanfaatkan kelemahan pada sistem kunci. Pengujian pada proses enkripsi juga menunjukkan adanya hubungan yang jelas antara ukuran *file* dan waktu pemrosesan, di mana semakin besar ukuran *file* yang dienkripsi, semakin lama pula waktu yang dibutuhkan untuk menyelesaikan proses tersebut. Hal ini menunjukkan bahwa performa aplikasi sangat dipengaruhi oleh kompleksitas data yang diolah, namun tetap berada pada batas waktu yang wajar untuk digunakan secara praktis.

4. Kesimpulan

Penelitian ini berhasil menjawab permasalahan lemahnya keamanan penyimpanan *file* data pelanggan pada toko *online By Barra.Outfit*. Dengan menerapkan kombinasi *Vigenere Cipher* dan RC4 Modifikasi (multiple KSA dan *drop* 768 *byte*), aplikasi berbasis *web* yang dikembangkan mampu melindungi *file* *Excel* (.xlsx) berisi informasi pelanggan sehingga tidak dapat dibaca tanpa kunci yang benar.

Pengujian fungsionalitas menunjukkan seluruh fitur berjalan sesuai rencana, pengujian keamanan membuktikan adanya sensitivitas tinggi terhadap perubahan kunci, dan pengujian performa menunjukkan waktu Proses enkripsi tercepat 0.07 detik untuk *file* 11KB dan terlama 2.45 detik untuk *file* 9 MB. Dengan demikian, kombinasi algoritma *Vigenere Cipher* dengan RC4 modifikasi terbukti meningkatkan keamanan *file* statis pelanggan pada *e-commerce* dengan waktu yang relatif singkat. Adapaun saran penelitian lanjutan dapat dilakukan dengan :

a. menambahan fitur keamanan tambahan
b. Optimasi kinerja untuk *file* lebih besar >10MB
c. Perbandingan dengan algoritma moderen
d. Pengujian lebih lanjut dengan randomness test

DAFTAR PUSTAKA

[1] N. Y. Setyawati, A. N. Khofid, A. U. . Rund, and V. Wati, "Modifikasi Kriptografi Klasik Kombinasi Metode *Vigenere Cipher* dan Caesar Cipher (Modification of Classical Cryptography Combination of the *Vigenere Cipher* and Caesar Cipher Methods)," *J. Smart Syst.*, vol. 1, no. 1, pp. 1–8, 2021.

[2] D. Shadani, A. I. Fahrozi, H. Fahriza, R. Khaliq, and R. Alpiansyah, "Peran Teknologi Informasi dalam Melindungi Privasi Pelanggan terhadap Keamanan Data dalam *E-commerce*," *J. Ilmu Komput. dan Inform. E-ISSN 3063-9026*, vol. 1, no. 3, pp. 51–54, 2025.

[3] T. A. Cahyaaty, I. Wijaya, muhamad Dafini Al Dzky, H. Gita Prasojo, and S. Hadi Prakoso, "Analisis Keamanan Data Pribadi Pada Pengguna *E-commerce* Dalam Mencegah Ancaman Data Pribadi," *JIFORTY*, vol. 5, no. 2, pp. 133–144, [*Online*]. Available: https://ojs.unimal.ac.id/index.php/joses/article/view/17121%0Ahttps://ojs.unimal.ac.id/index.php/joses/article/view *File* /17121/6676, 2024.

[4] A. P. R. Tarigan, P. S. Ramadhan, and K. Ibnutama, "Penerapan Kriptografi Untuk Pengamanan Data Penjualan Sepatu Dengan Metode AES (Advanced Encryption Standard)," *J. Cyber Tech*, vol. 5, no. 1, p. 26, doi: 10.53513/jct.v5i1.7851, 2023.

[5] A. P. Ramadhani, N. P. Tami, A. Lestari, V. Wati, and W. Wartono, "Keamanan Data dengan Super Enkripsi Kombinasi Vigenere dan Atbash Cipher," *J. Inf. Technol.*, vol. 4, no. 2, pp. 231–240, 2024.

[6] A. A. Bai'at, M. R. Fahlevvi, and W. Ariandi, "Metode Algoritma RC4 (Rivest Code 4) Untuk Pengamanan Database Transaksi Pada Glory *Digital* Sablon," *Explore*, vol. 13, no. 1, pp. 20–31, doi: 10.35200/ex.v13i1.33, 2023.

[7] S. K. Dewi, "Perbandingan Cryptography Klasik *Vigenere Cipher* Dengan Cryptography Modern RC4 Dalam Tingkat Keamanan Jaringan Komputer," *JoMMiT J. Multi Media dan IT*, vol. 8, no. 2, pp. 130–137, 2024.

[8] E. T. Rante, M. Alnando, M. Heru, and K. Junior, "Analisi Keamanan data Pelanggan Menggunakan Algoritma *Vigenere Cipher* dan Playfair," *J. SITEBA*, vol. 2, no. 1, pp. 24–29, 2023.

[9] M. F. Gifari, M. A. Dipani, S. Abdillah, and A. T. Zy, "Proses Enkripsi Berlapis Menggunakan Teknik Kriptografi Vignere dan Caesar," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 2, pp. 1959–1965, 2025.

[10] F. A. Hudi, S. Agustini, and M. Kurniawan, "Implementasi Kriptografi Enkripsi Pesan Dengan Metode Modifikasi (Initialization Vector) Algoritma RC4," *Pros. Semin. Nas. ...*, vol. 4, pp. 121–128, [*Online*]. Available: http://ejurnal.itats.ac.id/snestik/article/view/1769, 2021.