JURNAL ILMU KOMPUTER KHARISMA .TECH

Jurnal KHARISMA Tech | ISSN:1907-2317 | e-ISSN: 2810-0344
https://jurnal.kharisma.ac.id/kharismatech
published by: Pusat Penelitian STMIK KHARISMA Makassar

Volume: 20 no. 01 – Maret 2025- hlm. 44-56

Implementasi Algoritma RC4+ Pada Keamanan Sistem Komunikasi Chatting pada WEBSITE SAHEB

Oleh:

Jericho Anderson Kumala[1], Baizul Zaman[2*], Syamsul Bahri[3]
[1,2]Teknik Informatika, STMIK KHARISMA Makassar
e-mail: [1]jerichoanderson_21@kharisma.ac.id, [2]Baizul@kharisma.ac.id,
[3]Syamsul@kharisma.ac.id

Abstrak: *Keamanan data dalam komunikasi daring merupakan isu krusial, terutama dalam aplikasi yang menangani informasi sensitif seperti SAHEB, yang menyimpan data chat pribadi pengguna di basis data Firebase di Cloud. Penelitian ini bertujuan untuk mengimplementasikan algoritma RC4+ 256-bit pada data pengguna aplikasi SAHEB untuk meningkatkan keamanan data. Algoritma RC4+ dipilih karena menawarkan peningkatan keamanan dan efisiensi dibandingkan RC4. Pengumpulan data dilakukan melalui observasi dan pembuatan 10 data dummy chat pada webapp SAHEB yang mensimulasikan data pengguna. Implementasi algoritma RC4+ dilakukan menggunakan bahasa JavaScript. Hasil penelitian menunjukkan bahwa data chat berhasil dienkripsi dan didekripsi dengan waktu rata-rata enkripsi 0,30 ms dan dekripsi 0,19 ms. Selain itu, Avalanche Effect yang dihasilkan sebesar 3,009%, menunjukkan efektivitas algoritma dalam menjaga integritas data.*

Kata kunci: *Keamanan Data,Komunikasi Daring,Algoritma RC4+ Aplikasi SAHEB,Firebase,JavaScript,Avalanche Effect,Informasi Sensitif,Data Chat Pribadi.*

Abstract: *Data security in online communication is a crucial issue, especially in applications that handle sensitive information such as SAHEB, which stores users' private chat data in a Firebase database in the Cloud. This study aims to implement the RC4+ 256-bit algorithm on SAHEB application user data to enhance data security. The RC4+ algorithm was chosen for its improved security and efficiency over the RC4 algorithm. Data was collected through observation and creation of 10 dummy chat data sets on the SAHEB web app, simulating user data. The implementation of the RC4+ algorithm was conducted using JavaScript. The study results show that the user chat data was successfully encrypted and decrypted, with an average encryption time of 0.30 ms and an average decryption time of 0.19 ms. Additionally, the Avalanche Effect produced was 3.009%, demonstrating the algorithm's effectiveness in maintaining data integrity.*

Keywords: *Data Security,Online Communication,RC4+ 256-bit Algorithm,SAHEB Application,Firebase,JavaScript,Avalanche Effect,Sensitive Information,Private Chat Data*

1. Pendahuluan

Perkembangan teknologi informasi telah membawa dampak signifikan, termasuk dalam layanan kesehatan hewan yang kini semakin mudah diakses secara daring. Website SAHEB (Sahabat Hewan Bahagia) menyediakan platform konsultasi kesehatan hewan secara online, namun menghadapi tantangan terkait keamanan data komunikasi antara pengguna dan dokter hewan. Dalam upaya melindungi informasi sensitif, seperti data pribadi pemilik hewan dan rekam medis hewan peliharaan, penerapan enkripsi menjadi krusial. Algoritma RC4+,

* Corresponding author : Baizul Zaman (@kharisma.ac.id)

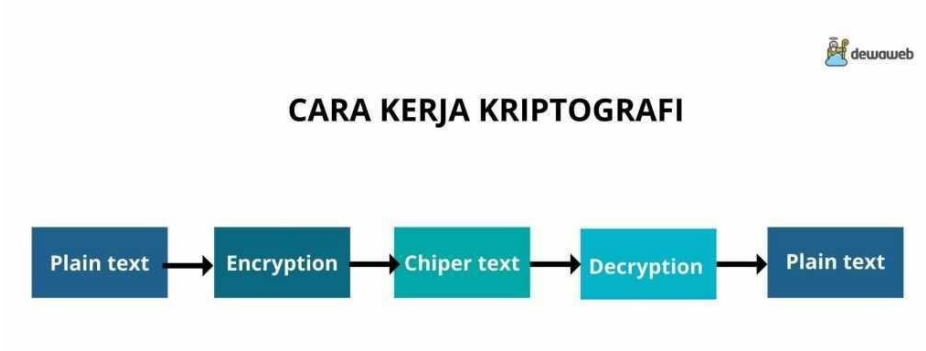
Diterima : *September, 2024* ***Disetujui :*** *November, 2024* ***Dipublikasikan :*** *April, 2025*

sebagai peningkatan dari RC4, menawarkan keamanan lebih baik dan efisiensi dalam proses enkripsi dan dekripsi, menjadikannya pilihan tepat untuk melindungi data komunikasi real-time dalam layanan chatting SAHEB. Dengan implementasi ini, SAHEB berkomitmen untuk meningkatkan keamanan data pengguna dan memastikan pengalaman layanan yang aman dan terpercaya. Tujuan dan alasan utama dari implementasi algoritma RC4+ pada sistem komunikasi chatting di website SAHEB adalah untuk memastikan bahwa semua data komunikasi antara pengguna dan dokter hewan terlindungi dari akses tidak sah dan kebocoran data. Hasil dan Pembahasan; pembahasan data dan analisa terkait metode yang digunakan. Berdasarkan latar belakang diatas rumusan masalah dari latar belakang diatas adalah Bagaimana mengimplementasikan algoritma RC4+ untuk mengenkripsi dan dekripsi pesan pengguna aplikasi SAHEB di data base firebase.

2. METODE PENELITIAN

2.1 Landasan Teori

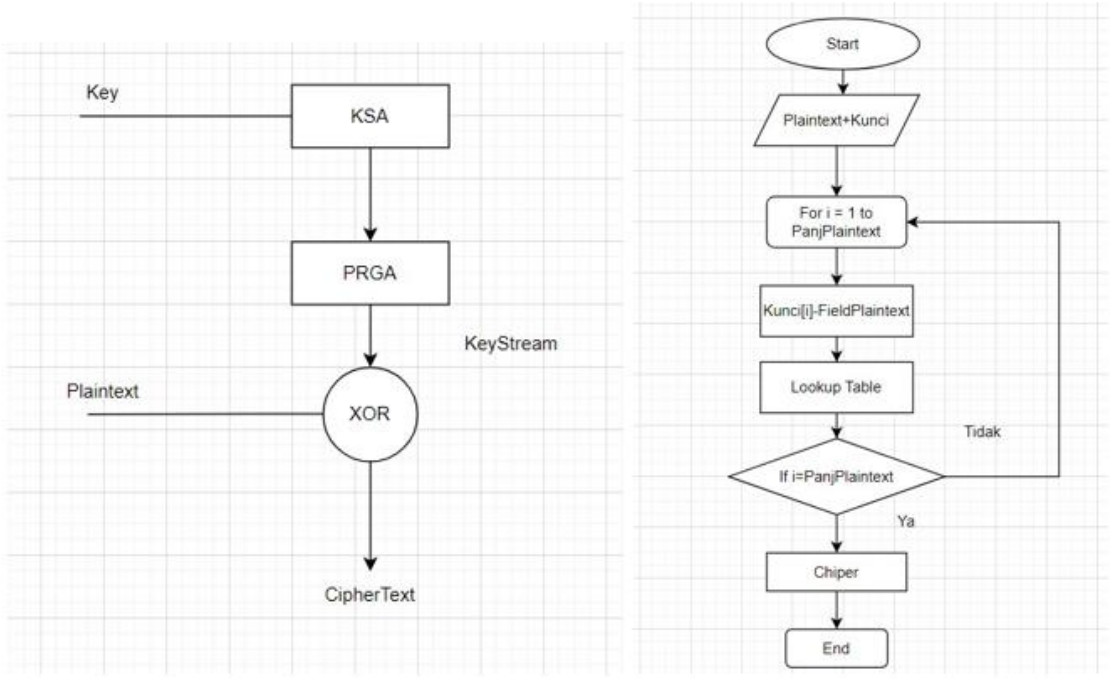
2.1.1 Kriptografi



Gambar 1 Cara Kerja Kriptografi

Kriptografi[3] Kriptografi berasal dari Bahasa Yunani: “cryptós” yang berarti “secret” (rahasia), dan “gráphein” yang berarti “writing” (tulisan). Oleh karena itu, kriptografi dapat diartikan sebagai “secret writing” (tulisan rahasia). Beberapa definisi kriptografi telah diajukan dalam berbagai literatur. Definisi yang umumnya digunakan sebelum tahun 1980-an menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Meskipun definisi ini sesuai dengan penggunaan kriptografi pada masa lalu, terutama dalam keamanan komunikasi militer, diplomatik, dan mata-mata, kini kriptografi memiliki peran lebih luas. Saat ini, kriptografi tidak hanya berkaitan dengan privasi, tetapi juga terkait dengan integritas data, otentikasi, dan nonrepudiation.

2.1.2 RC4+("Rivest Cipher 4 Plus")



Gambar 2 diatas menjelaskan tentang skema enkripsi dan dekripsi algoritma

[1], [3], [4], [5], [6], [7], [8], [9], [10]

RC4+ merupakan salah satu bentuk dari algoritma RC4, yang termasuk dalam kelompok algoritma kunci simetris dan berfungsi sebagai stream cipher. Dalam prosesnya, algoritma ini melakukan enkripsi dan dekripsi secara byte per byte dengan menggunakan kunci yang sama. Ini berarti setiap karakter atau bit plaintext diubah menjadi ciphertext secara bertahap, satu karakter atau satu bit setiap kali proses transformasi dilakukan.

Struktur dasar RC4+ mirip dengan algoritma RC4 standar, namun dengan penambahan beberapa operasi untuk meningkatkan keamanan cipher. Tujuannya adalah untuk memanfaatkan keunggulan dari algoritma RC4 sambil memberikan fitur tambahan yang mengatasi kelemahan yang ditemui dalam RC4, seperti probabilitas tinggi dalam penghasilan kekunci yang terprediksi.

Dalam algoritma RC4+, terdapat dua komponen utama, yaitu Key Scheduling Algorithm (KSA) dan Pseudo Random Generation Algorithm (PRGA). Kedua komponen ini bekerja sama untuk menghasilkan kunci yang digunakan dalam proses enkripsi dan dekripsi, serta untuk menghasilkan aliran bit acak yang digunakan dalam proses pembangkitan keystream.

Penerapan algoritma RC4+ dalam aplikasi SAHEB bertujuan untuk meningkatkan keamanan komunikasi dengan menggunakan proses enkripsi dan dekripsi yang kuat dan efisien, memastikan bahwa data sensitif yang dikirimkan melalui aplikasi tersebut tetap terlindungi dari ancaman yang mungkin terjadi selama pengiriman. Berikut merupakan contoh algoritma rumus rc4+

1. Inisialisasi KSA(Key Scheduling Algorithm): Input: Kunci k dengan panjang l bytes (0 ≤ l ≤ 256)
   Output: Array permutasi S

   for i from 0 to 255
   S[i] := i
   endfor

   j := 0 for i
   from 0 to 255
   j := (j + S[i] + k[i mod l]) mod 256
   swap S[i] and S[j]
   endfor
2. Inisiasi PRGA(Pseudo Random Generation Algorithm)
   Output: Keystream

   i :=
   0 j
   :=
   0
   while generating keystream
   bytes i := (i + 1) mod 256
   j := (j + S[i]) mod 256
   swap S[i] and S[j]
   K := S[(S[i] + S[j]) mod
   256] output K endwhile

2.1.3 JavaScript

JavaScript adalah bahasa pemrograman yang digunakan untuk mengontrol dan membuat konten interaktif di halaman web. Dikembangkan oleh Netscape, JavaScript memungkinkan pembuatan halaman web responsif dan dinamis melalui manipulasi DOM (Document Object Model). Selain di sisi klien, JavaScript juga digunakan di sisi server melalui platform seperti Node.js, membuatnya sangat populer dalam pengembangan web.

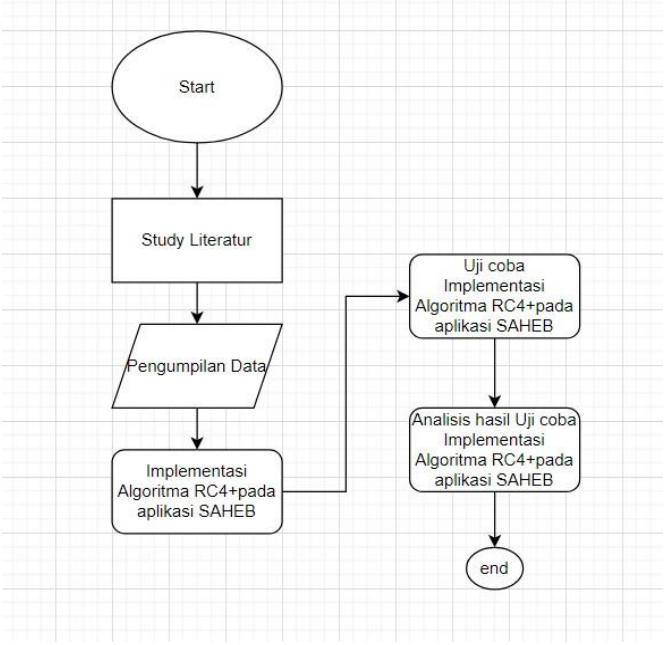
2.1.4 Nextjs

Next.js adalah kerangka kerja pengembangan web open-source berbasis JavaScript yang fokus pada konsep React. Mendukung Server-side Rendering (SSR) dan Static Site Generation (SSG), Next.js menyediakan routing otomatis, Hot Module Replacement (HMR),

dan integrasi mudah dengan API. Digunakan untuk membangun aplikasi web responsif dengan performa tinggi dan manajemen state yang fleksibel.

2.1.5 Tahapan Penelitian

Pada Gambar 3 dapat dilihat tahapan penelitian yang terdapat pada penelitian ini.



Gambar 3 Tahapan Penelitian

1. Studi Literatur

Tahap pertama dalam penelitian ini adalah melakukan studi literatur terkait algoritma RC4. Studi literatur ini bertujuan untuk memahami konsep dasar dan prinsip kerja dari algoritma tersebut.

2. Pengumpulan Data

Pada tahap ini, penulis mengumpulkan data yang diperlukan untuk penelitian.

3. Implementasi Algoritma RC4 pada Aplikasi SAHEB

Pada tahap ini penulis mengimplementasi algoritma RC4 pada aplikasi SAHEB. Penulis akan menggunakan bahasa pemrograman yang sesuai untuk melakukan pengembangan algoritma RC4.

4. Uji Coba Implementasi Algoritma RC4 pada Aplikasi SAHEB

Tahap ini dilakukan untuk melakukan uji coba terhadap implementasi algoritma RC4 pada aplikasi SAHEB. Uji coba dilakukan untuk memastikan bahwa implementasi algoritma RC4 dapat berjalan dengan baik pada aplikasi SAHEB dan efektif.

5. Analisis Hasil Uji Coba Implementasi Algoritma RC4 pada Aplikasi SAHEB

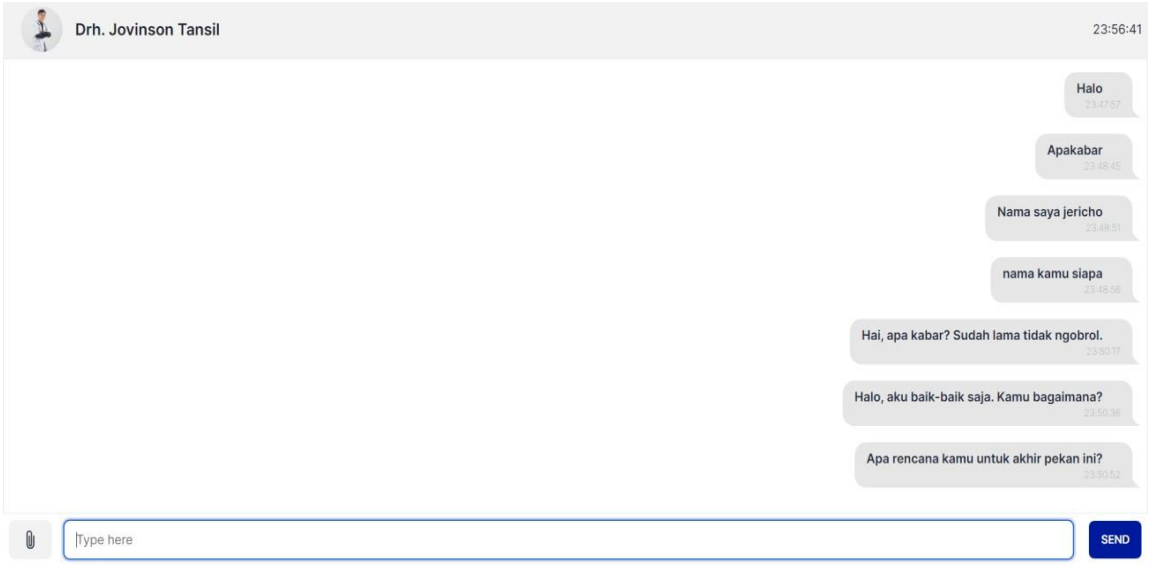
Setelah melakukan uji coba pada tahap sebelumnya, tahap selanjutnya adalah melakukan analisis terhadap hasil uji coba tersebut. Analisis ini bertujuan untuk mengevaluasi performa dan efektivitas algoritma RC4 yang telah diimplementasikan.

6. Kesimpulan

Menyimpulkan temuan dan hasil dari penelitian ini.

2.1.6 Jenis & Sumber Data dummy

Penelitian ini memanfaatkan informasi yang diperoleh dari percakapan yang terjadi di platform SAHEB, di mana peneliti akan membuat 10 data dummy yang akan dienkripsi dan dekripsi menggunakan algoritma RC4+. Pada Gambar 4 dapat dilihat data dummy yang akan digunakan pada penelitian ini.



Gambar 3 Data dummy yang akan digunakan untuk menguji

Table 1 Berikut 10 data dummy yang akan penulis gunakan untuk menguji

NO	Pesan Plaintext	Pesan Cipher(hasil enkripsi)
1	Halo	
2	Apakabar	
3	Nama saya jericho	
4	nama kamu siapa	
5	Hai, apa kabar? Sudah lama tidak ngobrol.	
6	Halo, aku baik-baik saja. Kamu bagaimana?	
7	Apa rencana kamu untuk akhir pekan ini?	
8	Hai, bisa berikan nomor pesanan Anda?	
9	Baik, saya akan cek dulu. Mohon tunggu sebentar.	
10	Bagian mana yang belum kamu mengerti?	

Data dummy pada Gambar 4 merupakan plaintext yang akan dilakukan proses enkripsi dan dekripsi. Selain pengujian enkripsi dan dekripsi plaintext, peneliti juga melakukan evaluasi waktu yang dibutuhkan untuk proses enkripsi dan dekripsi. Dengan membuat timer pada aplikasi copy SAHEB, dengan tujuan mencari tahu seberapa cepat proses enkripsi pada

algoritma RC4+ 256bit. Timer yang dibuat menggunakan metode bawaan bahasa Javascript yaitu, *nanoTime()* dan memiliki satuan milisekon(ms).

3. HASIL DAN PEMBAHASAN

3.1. Proses Enkripsi dan Dekripsi

Peneliti melakukan enkripsi dan dekripsi pada data dummy di gambar 2 dan hasil yang didapatkan adalah:

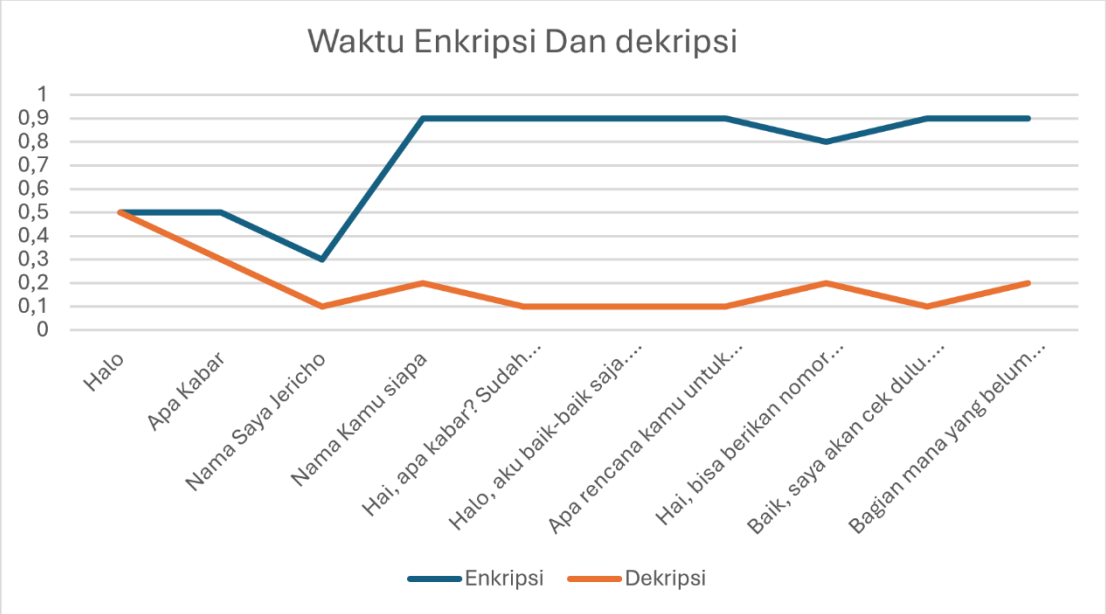
Table 2 Hasil Enkripsi dari Dari 10 Plaintext data dummy

NO	Pesan Plaintext	Pesan Cipher(hasil enkripsi)
1	Halo	Óhœ×
2	Apakabar	ÚyœÓ_½Òi
3	Nama saya jericho	ÕhœÙ-¬ÒbœÄtœLoœiÃ`
4	nama kamu siapa	õhœÙ-´ÒvœÄmœ_vœ
5	Hai, apa kabar? Sudah lama tidak ngobrol.	Óhœœ-¾ÃzÌœ□œ_tÁ!ÿ5É\þíœqóÖÝÑœ¹ÍéÌh<œrœB}F
6	Halo, aku baik-baik saja. Kamu bagaimana?	Óhœ×œÿÒpœÄ|œWmÓcÍ)Æœå¬œq°œ¶Äœ¨œàœa:-}œCpW
7	Apa rencana kamu untuk akhir pekan ini?	ÚyœœLºÝxœœ□ÑUgœtœ5ÃIã¦^qõßœ×È-Ééœh{-~œœ
8	Hai, bisa berikan nomor pesanan Anda?	Óhœœ-½ÚhœÄ|œLœ`Â`ÃRû¢0îÒœÄœ¼Â¢-h?œ/
9	Baik, saya akan cek dulu. Mohon tunggu sebentar.	ÙhœÓœÿÀzœœ>œUgœ!Ï%Æœò¸œe°œ°Êœ²Â¢œs5œwœbö£|œñ¸
10	Bagian mana yang belum kamu mengerti?	ÙhœÑ_±œvœœ□ÑGgœfœ"ÈQã ^{ÿÚœœœ¸Âåœt/-/

Algoritma RC4+ bekerja dengan menambahkan beberapa langkah pada algoritma RC4 standar, seperti swapping tambahan dan operasi XOR tambahan untuk meningkatkan keamanan. Proses enkripsi dan dekripsi dilakukan dengan menginisialisasi state array berdasarkan kunci, menghasilkan keystream, dan menggunakan keystream ini untuk mengenkripsi atau mendekripsi data. Padding ditambahkan untuk memastikan panjang blok sesuai, dan dihapus setelah dekripsi.

3.2 Evaluasi Waktu Proses Enkripsi

Peneliti melakukan evaluasi waktu proses enkripsi pada aplikasi tiruan SAHEB yang telah diterapkan algoritma RC4+. Dengan memanfaatkan metode *nanoTime*() untuk kalkulasi waktu enkripsi. Data dummy yang dibuat pada tabel 1 akan digunakan dan menggunakan kunci yang sama untuk mengukur kecepatan enkripsi dan dekripsi algoritma RC4+ 256-bit.



Gambar 4 Waktu Enkripsi Dan Dekripsi

Waktu yang dibutuhkan untuk proses enkripsi sebagian besar stabil dengan ratarata waktu sekitar 0,5 milidetik, kecuali pada teks yang lebih pendek ("Nama Saya Jericho" dan "Nama Kamu siapa") yang memiliki waktu enkripsi lebih rendah (0,3 milidetik dan 0,9 milidetik).

Waktu dekripsi bervariasi lebih banyak dibandingkan waktu enkripsi. Dekripsi pada beberapa teks panjang memerlukan waktu yang lebih sedikit (0,1 milidetik) dibandingkan dengan teks pendek (0,5 milidetik untuk "Halo").

Algoritma RC4+ menunjukkan efisiensi yang baik dalam waktu enkripsi dengan hasil yang konsisten. Namun, efisiensi dekripsi tampaknya dipengaruhi oleh panjang teks yang dienkripsi.

3.3. Pengujian Avalanche Effect

Pengujian Avalanche Effect dilakukan untuk mengetahui tingkat keacakan dari hasil enkripsi, dengan cara melakukan enkripsi pada plaintext, lalu mengubah salah satu karakter dari plaintext tersebut dan dilakukan proses enkripsi ulang. Setelah itu, ciphertext yang dihasilkan akan diubah menjadi biner, dan dibandingkan. Selanjutnya, hasil perbandingan akan dimasukkan ke dalam rumus AE (Avalanche Effect). Data yang akan digunakan untuk melakukan pengujian Avalanche Effect. Data yang akan digunakan untuk melakukan pengujian Avalanche Effect adalah data dummy yang telah dibuat di gambar 4 Pengujian dilakukan menggunakan algoritma RC4+ dengan kunci 256-bit dan kunci yang sama.

Berdasarkan pengujian, didapatkan hasil:

$$AE = \frac{Jumlah\ Bit\ Yang\ Berubah}{Total\ Jumlah\ Bit} \times 100\%$$

Table 3 Presesntase Hasil Tingkat Keacakan

No	Plaintext1	Plaintext2	Presentase AE(tingkat keacakan)
1	Halo	Hala	5.36%
2	Apakabar	Apakabae	3.85%
3	Nama saya jericho	Nama saya jerichi	4.04%
4	nama kamu siapa	Nama kamu siape	3.12%
5	Hai, apa kabar? Sudah lama tidak ngobrol.	Hai, apa kabar? Sudah lama tidak ngobror.	2.93%
6	Halo, aku baik-baik saja. Kamu bagaimana?	Halo, aku baik-baik saja. Kamu bagaimane?	3.32%
7	Apa rencana kamu untuk akhir pekan ini?	Apa rencana kamu untuk akhir pekan ino?	3.80%
8	Hai, bisa berikan nomor pesanan Anda?	Hai, bisa berikan nomor pesanan Andi?	0.21%
9	Baik, saya akan cek dulu. Mohon tunggu sebentar.	Baik, saya akan cek dulu. Mohon tunggu sebental.	2.21%
10	Bagian mana yang belum kamu mengerti?	Bagian mana yang belum kamu mengerta?	1.25%

Berdasarkan kriteria SAC (Strict Avalanche Criterion), hasil Avalanche Effect yang mendekati 50% menunjukkan tingkat keacakan yang baik. Namun, setelah melakukan pengujian, algoritma RC4+ menunjukkan tingkat keacakan yang sangat rendah dengan rata-rata hanya sebesar 3,009%. Peneliti juga melakukan pengujian dengan algoritma RSA yang bertujuan untuk membandingkan kedua algoritma ini dari segi kecepatan dan tingkat keacakan, dari hasil pengujian diatas dapat diambil kesimpulan sebagai berikut: Algoritma yang diuji menunjukkan tingkat keacakan yang rendah berdasarkan Avalanche Effect. Rata-rata tingkat keacakan hanya mencapai 2.909%, yang mengindikasikan bahwa output enkripsi tidak cukup acak dan berbeda signifikan saat terdapat perubahan kecil pada plaintext. Untuk aplikasi yang memerlukan keamanan tinggi, tingkat keacakan yang rendah ini mungkin menjadi titik lemah karena pola enkripsi lebih mudah ditebak atau dianalisis. Oleh karena itu, diperlukan algoritma yang lebih efektif dalam menghasilkan keacakan yang tinggi untuk meningkatkan keamanan data.

3.3 Perbandingan dengan algoritma enkripsi RSA

Kali ini penguji akan melakukan perbandingan tingkat kecepatan dan tingkat keacakan algoritma RC4+ dengan algoritma RSA (Rivest-Shamir-Adleman) dimana, RSA (Rivest-Shamir-Adleman) adalah salah satu algoritma kriptografi kunci publik yang paling populer dan digunakan secara luas. Algoritma ini digunakan untuk mengamankan komunikasi dan memastikan integritas data. Berikut adalah penjelasan singkat tentang bagaimana RSA bekerja dan implementasi sederhananya:

Penjelasan Singkat RSA

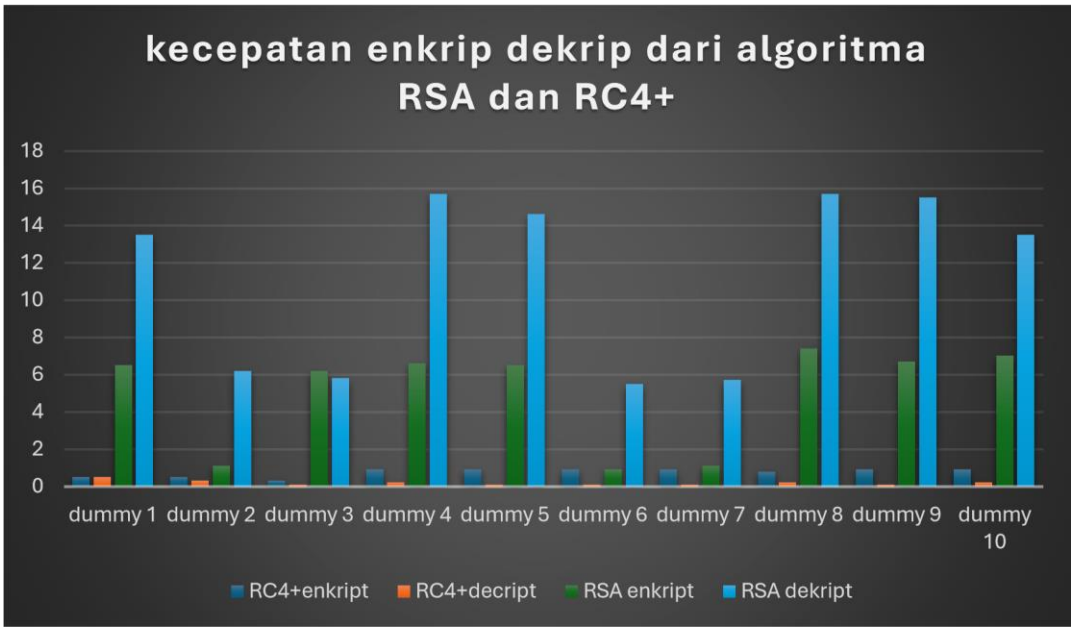
1. **Kunci Publik dan Privat**: o RSA menggunakan dua kunci yang berbeda: kunci publik untuk enkripsi dan kunci privat untuk dekripsi. o Kunci publik dapat dibagikan kepada siapa saja, sementara kunci privat harus dijaga kerahasiaannya.

2. Proses Pembuatan Kunci:

- Pilih dua bilangan prima besar, ppp dan qqq.
- Hitung nnn = p×qp \times qp×q. Nilai nnn digunakan sebagai modulus untuk kunci publik dan privat.
- Hitung nilai Euler's totient function, φ(n)\varphi(n)φ(n) = (p−1)×(q−1)(p-1) \times (q-1)(p−1)×(q−1).

1.1 Pilih bilangan bulat eee yang merupakan coprime dengan φ(n)\varphi(n)φ(n) dan 1<e<φ(n)1 < e < \varphi(n)1<e<φ(n). Nilai eee ini adalah eksponen kunci publik. o Hitung ddd sebagai kebalikan modular dari eee modulo φ(n)\varphi(n)φ(n). d×e≡1(modφ(n))d \times e \equiv 1 \pmod{\varphi(n)}d×e≡1(modφ(n)). Nilai ddd ini adalah eksponen kunci privat.

A. Perbandingan Kecepatan Algoritma RC4+ dan Algoritma RSA



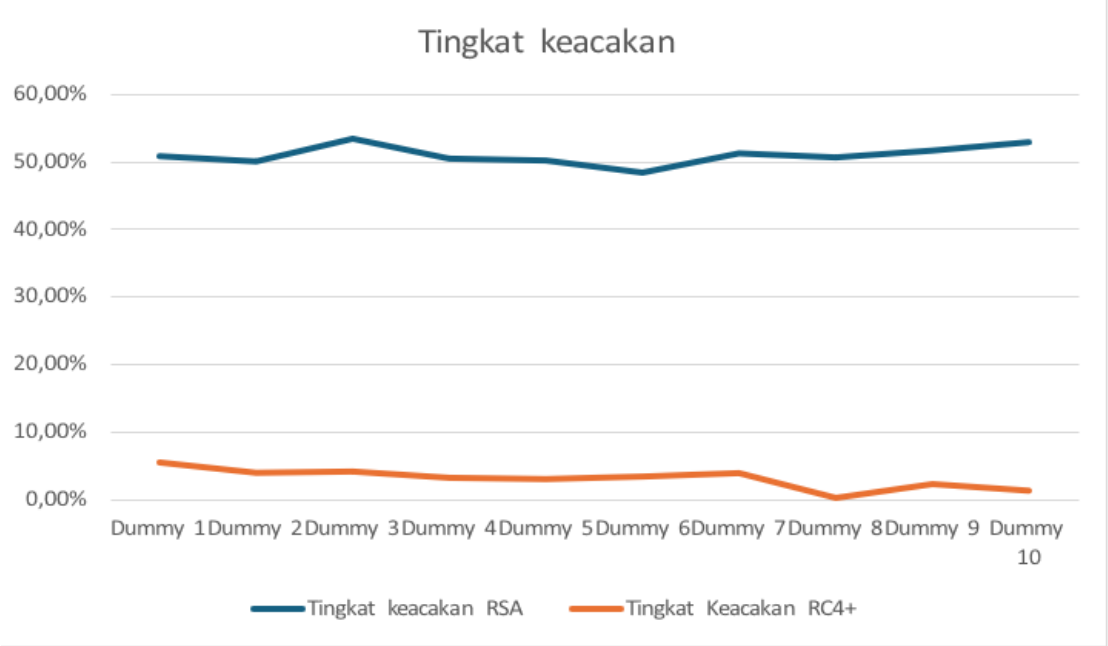
Gambar 5 Perbandingan waktu Enkripsi Dekripsi Algoritma RC4+ & RSA

RC4+: Algoritma ini sangat cepat baik dalam proses enkripsi maupun dekripsi, dengan waktu yang konsisten rendah.

RSA: Algoritma ini secara signifikan lebih lambat baik dalam proses enkripsi maupun dekripsi dibandingkan dengan RC4+. Proses dekripsi terutama memakan waktu lebih lama dibandingkan dengan enkripsi.

Secara keseluruhan, RC4+ lebih efisien dalam hal kecepatan dibandingkan dengan RSA, baik untuk enkripsi maupun dekripsi.

B. Perbandingan Tingkat Keacakan Algoritma RC4+ dengan Algoritma RSA



Gambar 6 Perbandingan tingkat keacakan Algoritma RC4+ & RSA

RSA: Algoritma ini memiliki tingkat keacakan yang jauh lebih tinggi, menunjukkan keefektifan dalam menghasilkan output yang sangat berbeda bahkan untuk plaintext yang hampir serupa. Ini menunjukkan kekuatan dalam hal keamanan dan ketidakmungkinan untuk menebak pola dari output yang dihasilkan.

RC4+: Algoritma ini memiliki tingkat keacakan yang lebih rendah dibandingkan dengan RSA. Meskipun lebih cepat, RC4+ kurang efektif dalam menghasilkan output yang acak dan berbeda signifikan, yang bisa menjadi kelemahan dari segi keamanan.

Secara keseluruhan, meskipun RC4+ lebih cepat dalam proses enkripsi dan dekripsi, RSA lebih unggul dalam hal tingkat keacakan dan keamanan enkripsinya.

4. KESIMPULAN

Berdasarkan evaluasi yang telah dilakukan, Aplikasi SAHEB dapat melakukan enkripsi dan dekripsi data chat pengguna dan Dokter. Selain itu hasil evaluasi performansi Waktu yang dibutuhkan untuk proses enkripsi sebagian besar stabil dengan rata-rata waktu 0,71 ms dan waktu rata-rata dekripsi yaitu 0,22 ms Waktu dekripsi bervariasi lebih banyak

dibandingkan waktu enkripsi. Dekripsi pada beberapa teks panjang memerlukan waktu yang lebih sedikit (0,1 milidetik) dibandingkan dengan teks pendek (0,5 milidetik untuk "Halo"). Algoritma RC4+ menunjukkan efisiensi yang baik dalam waktu enkripsi dengan hasil yang konsisten. Namun, efisiensi dekripsi tampaknya dipengaruhi oleh panjang teks yang dienkripsi. Evaluasi Avalanche Effect dilakukan untuk mengukur tingkat keacakan dari hasil enkripsi dari algortima RC4+256bit yang diimplementasikan, dan didapatkan hasil rata-rata presentasi 2.909%. Berdasarkan dari beberapa penelitian sebelumnya, nilai presentasi AE yang baik adalah 45%-50%.Algoritma yang digunakan dalam pengujian ini menunjukkan tingkat keacakan yang relatif stabil di kisaran yang rendah. Hal ini mengindikasikan bahwa algoritma tersebut mungkin kurang efektif dalam menghasilkan output yang benar-benar acak dan berbeda signifikan.dan penulis juga melakukan perbandingan waktu dan keacakan algoritma RC4+ dengan RSA (Rivest-Shamir-Adleman) dimana penguji menemukan keunggulan dan kekurangan masing-masing algoritma yaitu RC4+ lebih cepat dalam proses enkripsi dan dekripsi,sedangkan RSA lebih unggul dalam hal tingkat keacakan dan keamanan enkripsinya. Algoritma RC4+256bit dapat diimplementasikan pada aplikasi SAHEB yang menggunakan bahasa pemrograman JavaScript dengan menggunakan framework Next.js. Sehingga dari kedua pembahasan tersebut disimpulkan bahwa penggunaan kriptografi Algoritma RC4+256bit cocok digunakan untuk pengamanan data chat pengguna pada aplikasi sosial media seperti Aplikasi SAHEB.

DAFTAR PUSTAKA

[1] S. An-Nissa, H. Mawengkang, and S. Efendi, “RC4 GGHN Cryptography Algorithm for Message Security,” vol. 6, no. 2, 2022, doi: 10.30743/infotekjar.v6i2.4531.

[2] Z. Lubis and C. F. Sianturi, “Informasi dan Teknologi Ilmiah (INTI),” 2023.

[3] L. Oktasari, “BULLETIN OF COMPUTER SCIENCE RESEARCH Perbandingan Algoritma RC4+ dan RC4a dalam Pengamanan File Teks”, [Online]. Available: https://hostjournals.com/bulletincsr

[4] O. Zalukhu, “Implementasi Algoritma RC4+ dan Metode Spread Spectrum Untuk Mengamankan Pesan Rahasia Kedalam Citra Digital,” 2021.

[5] D. Iqbal, A. R. Panggabean, I. W. Sinaga, and T. Zebua, *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) Implementasi Algoritma RC4+ Untuk Mengamankan Pesan Teks Pada Aplikasi Chatting*. [Online]. Available: https://seminar-id.com/semnassainteks2019.html

[6] N. C. Lase, “Pengamanan Teks Terenkripsi Dengan Algoritma RC4+ Dan Steganografi DCS Pada Citra Digital”.

[7] Z. Khoiriah Siregar and A. Ulva, *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) Pengamanan Audio Menggunakan Algoritma RC4+*. [Online]. Available: https://seminar-id.com/semnas-sainteks2019.html

[8] M. A. Budiman, D. Rachmawati, and R. A. Badegeil, “The implementation of RC4+ and Variably Modified Permutation Composition algorithms in the three-pass protocol scheme for data security,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2019. doi: 10.1088/1742-6596/1235/1/012085.

[9] M. A. Budiman, Amalia, and N. I. Chayanie, “An Implementation of RC4+ Algorithm and Zig-zag Algorithm in a Super Encryption Scheme for Text Security,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2018. doi: 10.1088/17426596/978/1/012086.

[10] S. Karo-Karo, Tulus, and M. Zarlis, “Analysis of the formation of a dynamic brief key algorithm RC4+ for file security,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1757-899X/725/1/012135.