Jurnal Ilmiah Ilmu Komputer
Banthayo Lo Komputer Vol. 4, No. 1 (2025)
E-ISSN : 2827-9425
P-ISSN : 2828-4666

PERBANDINGAN KINERJA ALGORITMA RSA DAN ALGORITMA RC4 DALAM MENGENKRIPSI DAN DEKRIPSI DATA FILE BERSASIS ANDROID

[1]Moh. Rezaldy D. Kasili, [2*]Hastuti Dalai, [3]Warid Yunus, [4]Zohrahatay
[1,2,3,4]Fakultas Ilmu Komputer, Taknik Informatika, Universitas Ichsan Gorontalo, Gorontalo, Indonesia
Email: [1]rezaldykasili123@gmail.com, [2*]hannadalai88@gmail.com, [3]warid.dsn@gmail.com, Zohrahayaty@gmail.com

Abstrak-Penelitian ini membandingkan kinerja algoritma RSA dan RC4 dalam mengenkripsi dan mendekripsi file data pada sistem berbasis Android. Tujuan utama adalah untuk mengevaluasi kecepatan dan keamanan kedua algoritma tersebut guna menentukan kesesuaiannya untuk berbagai aplikasi. Hasil penelitian menunjukkan bahwa RC4 cenderung lebih cepat dalam proses enkripsi dan dekripsi dibandingkan RSA. Perbedaan ini disebabkan oleh kompleksitas algoritma dan metode enkripsi yang digunakan. Namun, RSA umumnya dianggap lebih aman daripada RC4, karena RC4 telah terkena serangan kriptanalisis yang serius. RSA menggunakan kunci publik dan kunci pribadi, sedangkan RC4 menggunakan kunci simetris. RSA sering digunakan untuk aplikasi yang membutuhkan tingkat keamanan tinggi, seperti dalam pengamanan komunikasi melalui jaringan. Meskipun cepat, RC4 telah banyak ditinggalkan dalam banyak aplikasi karena kelemahan keamanannya. Dengan demikian, pemilihan antara RSA dan RC4 tergantung pada kebutuhan spesifik pengguna, dengan pertimbangan utama antara keamanan dan kinerja.

Kata kunci: algoritma, RSA, RC4, enkripsi data, dekripsi data, Android

Abstract *This research compares the performance of RSA and RC4 algorithms in encrypting and decrypting data files on Android-based systems. The main objective is to evaluate the speed and security of the two algorithms to determine their suitability for various applications. The results show that RC4 tends to be faster in encryption and decryption than RSA. The difference is due to the algorithm complexity and the encryption method used. However, RSA is generally considered more secure than RC4, as RC4 has been subject to serious cryptanalysis attacks. RSA uses public keys and private keys, while RC4 uses symmetric keys. RSA is often used for applications requiring a high security level such as securing communications over a network. Although fast, RC4 has been widely abandoned in many applications due to its security weaknesses. Thus, the choice between RSA and RC4 depends on the specific needs of the user. The major consideration is about security and performance.*

Keywords*: algorithm, RSA, RC4, data encryption, data decryption, Android*

1. PENDAHULUAN

Dengan perkembangan teknologi yang sangat pesat, kita telah mengalami perkembangan informasi apa pun dalam bidang pendidikan maupun bidang lainnya. Contoh perkembangan tersebut adalah internet, yang saat ini memungkinkan banyak orang bertukar data secara bebas melalui jaringan. Berkat kenyamanannya, internet telah menjadi salah satu media masa paling populer di dunia[1]. Keamanan data adalah salah satu hal yang patut mendapat perhatian lebih, terutama bagi pengguna yang selalu melakukan proses pertukaran data rahasia, sehingga perlu dilakukan pengamanan data agar beberapa pihak yang tidak memiliki kewenangan tidak dapat membuka informasi yang di kirimkan[2].

Android adalah sistem operasi yang cukup rentan terhadap serangan *malwere,* android sering terhubung dengan jaringan nirkabel publik atau *hot-spot,* yang sering kali tidak aman. Penyerangan dapat menggunakan jaringan ini untuk mencuri data file atau melakukan serangan yang membahayakan keamanan data file. Oleh karena itu dibutuhkan algoritma kriptografi dalam pengamanan data file berbasis android. Android adalah platform perangkat sumber terbuka, karena itu android dirancang untuk bekerja di semua jenis perangkat[3].

Kriptografi adalah ilmu sekaligus seni keamanan informasi. Kriptografi ada dua proses utama yaitu proses penyandian pesan yang dapat dibaca (*plaintext*) menjadi pesan yang tidak dapat dibaca atau telah disandikan (*chipertext*) disebut dengan proses enkripsi, dan proses pengembalian *chipertext* ke *plaintext* disebut dengan dekripsi[1]. Ada beberapa macam algoritma kriptografi dalam pengamanan data seperti algoritma Rivest Shamir Adler (RSA) dan algoritma Rivest *Cod*e 4 (RC4).

Algoritma Rivest Shamir Adler (RSA) adalah algoritma enkripsi kunci publik. RSA adalah algoritma

E-ISSN : 2827-9425
P-ISSN : 2828-4666

pertama yang diketahui paling cocok untuk menandai dan enkripsi, dan salah satu penemuan besar pertama dalam kriptografi kunci publik[4]. Algoritma ini memiliki keamanan yang terletak pada kompleksitas komputasi logaritma diskrit[1].

Algoritma kriptografi Rivest Code 4 (RC4) adalah salah satu algoritma kunci simetris yang dibuat oleh RSA *Dara Security Inc* (RSADSI) dalam bentuk *stream chiper*. Algoritma ini ditemukan pada tahun 1987 oleh Ronald dan Rivest dan menjadi simbol keamanan RSA. RC4 menggunakan kunci dengan panjang antara 1 dan 256 byte yang untuk menginisialisasi tabel sepanjang 256 byte[5].

2. TINJAUAN PUSTAKA

2.1 Perbandingan

Perbandingan adalah proses membandingkan atau mengevaluasi dua atau lebih hal untuk menentukan perbedaan, kesamaan, atau hubungan diantara mereka. Ini melibatkan memeriksa karakteristik, sifat, atau kualitas dari benda atau konsep yang dibandingkan. Perbandingan sering digunakan untuk memahami properti relatif dari objek dan membuat penilaian atau keputusan berdasarkan perbedaan atau kesamaan yang di temukan.

2.2 Kinerja Kriptografi

Kinerja algoritma khususnya algoritma kriptografi adalah serangkaian aturan matematis yang digunakan untuk mengamankan komunikasi dan data. Kinerjanya dapat dijelaskan melalui beberapa aspek yaitu [5]: keamanan, kecepatan, efisiensi, kunci, dan tahan terhadap seranagan.

2.3 Kriptografi

Kriptografi berasal dari bahasa yunani yaitu "*The Art Of Secreet Writing*" yang berarti seni menulis dengan rahasia, yang tujuannya adalah untuk membuat pesan menjadi tidak ada arti[6][7]. Kriptografi memiliki dua konsep dasar yaitu enkripsi dan dekripsi, enkripsi adalah proses melindungi pesan asli (*p()laintext*) dengan mengubah bentuk data menjadi kode menggunakan kunci yang telah ditentukan oleh pengirim dan penerima pesan menjadi pesan rahasia (*chipertext*) sehingga data atau informasi yang dikirimkan tidak dapat dibaca sehingga dapat dirahasiakan. Sedangkan dekripsi adalah proses pembalikan pesan dalam bentuk (*chipertext*) menjadi pesan aslinya (*plaintext*)[6][8]. Kriptografi dapat dibagi menjadi dua kategori utama berdasarkan cara penggunaan kunci :

2.4 Algoritma RSA

Algoritma RSA pertama kali dikembangkan oleh Ron Rivest, Adi Shamir, dan Led Adlenan dari Massachusetts Institute of Thechnology pada tahun 1978. Nama RSA sendiri diambil dari tiga nama peneliti, yaitu:

(R)ivest, (S)hamir, dan (A)dleman. RSA adalah algoritma kunci publik yang memiliki dua kunci, kunci publik (*public key*) dan kunci pribadi (*private key*). RSA dibagi menjadi tiga proses yaitu pembuatan kunci, enkripsi, dan dekripsi. Proses enkripsi dan dekripsi dilakukan dengan konsep bilangan prima dan modulo aritmatika. Kunci enkripsi tidak dirahasiakan dan diberikan kepada publik sedangkan kunci dekripsi bersifat rahasia, untuk menemukan kunci dekripsi dilakukan dengan menguraikan bilangan bulat menjadi pengganda keunggulan[1]. Besaran-besaran yang digunakan pada algoritma RSA antara lain:

1. p dan q bilangan prima (rahasia)
2. $n = pq$ (tidak rahasia)
3. $p(n) = (p\text{-}1) (p\text{-}1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plaintext) (rahasia)
7. c (chipertext) (tidak rahasia)

2.5 Algoritma RC4

RC4 adalah jenis *stream chiper* yang memproses unit pengukuran atau unit data dalam satu waktu. Dengan demikian, enkripsi dan dekripsi dapat di implementasikan dengan panjang yang variabel. Algoritma

E-ISSN : 2827-9425
P-ISSN : 2828-4666

ini tidak perlu menunggu sejumlah input data sebelum memproses atau menambahkan byte tambahan untuk enkripsi[9].

Penerapan RC4 secara manual melibatkan langkah-langkah untuk menghasilkan *keystream* dan melakukan XOR dengan teks terbuka (*plaintext*). Berikut adalah contoh langkah-langkah secara manual:

1. Inisialisasi S-box (*State Box*)
   - Buat array S-box berisi nilai 0 hingga 255.
   - Misalnya S = [0,1,2,….,255].
2. Inisialisasi KSA (*Key Scheduling Algorithm*)
   - Gunakan kunci (Key) sebagai input.
   - Permutasikan nilai dalam S-box berdasarkan kunci.
   - Langkah ini melibatkan pertukaran elemen-elemen S-box sesuai dengan Algoritma RC4.
3. Inisialisasi PRGA (*Pseudo-Random Generation Algorithm*):
   - Gunakan S-box yang telah diinisialisasi.
   - Iterasi PRGA sebanyak byte pada teks terbuka (plaintext).
   - Dalam setiap iterasi, pertukaran elemen S-box dan hasilkan byte dari *keystream.*
4. XOR dengan *Plaintext*
   - Dapatkan keystream dari PRGA.
   - Lakukan operasi XOR antara byte keystream dan byte
5. *plaintext* untuk menghasilkan byte *chipertext*

2.6 Enkripsi

Enkripsi adalah proses yang dilakukan untuk menyandikan teks biasa menjadi teks sandi sehingga pesan tidak dapat dibaca oleh orang yang tidak berwenang[9]. Enkripsi sangat penting dalam kriptografi yang menjamin keamanan data yang dikirimkan dan dirahasiakan. Pesan asli disebut *plaintext* yang diubah menjadi kode yang tidak bisa dipahami atau disebut dengan *chipertext*[10].

2.7 Dekripsi

Dekripsi adalah proses mengekstraksi tek biasa dari teks terenkripsi[9]. Dekripsi merupakan kebalikan dari enkripsi, pesan yang telah terenkripsi dikembalikan ke bentuk aslinya (*plaintext*) di sebut dekripsi pesan[10].

3. METODE PENELITIAN

3.1 Jenis Metode, Subjek, Waktu dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian teoritis. Metode ini menggunakan studi literatur. Oleh karena itu data penelitian ini diperoleh dengan memperbanyak literatur tentang algoritma RSA dan RC4 dibuku maupun jurnal penelitian sebelumnya.

3.2 Pengumpulan data

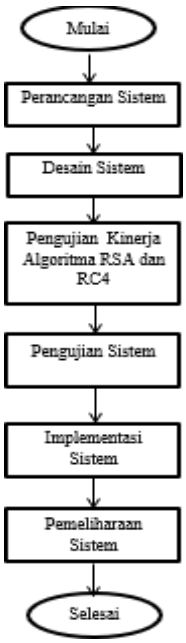
Metode pengumpulan data yang digunakan untuk mendapatkan data dan informasi digunakan hanya satu jenis data yaitu data sekunder. Data sekunder merupakan pengambilan informasi dengan melakukan pengkajian ke perpustakaan yang berisis dasar-dasar teori. Metode ini digunakan untuk mengambil contoh dokumen yang berhubungan dengan objek penelitian.

3.3 Pengembangan Sistem

3.3.1 Sistem yang diusulkan

Sistem yang diusulkan dapat digambarkan menggunakan *flowchart* dokumen yang pada gambar di bawah ini :

E-ISSN : 2827-9425
P-ISSN : 2828-4666



Gambar 1. Perancangan sistem

3.4 Pengujian Sistem

a. White Box

Software yang telah direkayasa kemudian diuji dengan metode *white box testing* pada kode program proses penerapan metodenya. Kode program tersebut di buatkan *flowchart* programnya, kemudian dipetakan ke dalam bentuk *flowgraph* yang tersusun dari beberapa *node* dan *edge*. Berdasarkan *flowgraph*, ditentukan jumlah *region* dan *cyclomatic complexity* (CC). apabila independent *path* = V(G)=(CC)=*Region* dimana setiap *path* hanya dieksekusi sekali dan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman.

b. ***Black box***

Selanjutnya *software* diuji pula dengan metode *blackbox* testing yang fokus pada keperluan fungsional dari *software* dan berusaha untuk menemukan kesalahan dalam beberapa kategori,diantaranya:

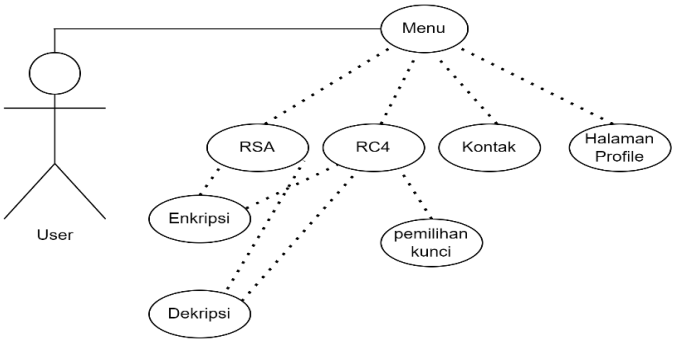
(1) Fungsi fungsi yang salah atau hilang;
(2) Kesalahan interface;
(3) Kesalahan dalam struktur data atau akses basis data eksternal;
(4) Kesalahan performa;
(5) Kesalahan inisialisasi dan terminasi.

4. HASIL DAN PEMBAHASAN

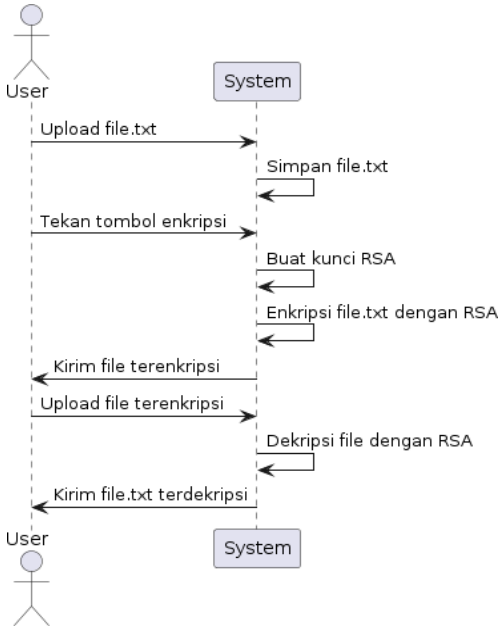
4.1 Perancangan Aplikasi

Untuk perancangan aplikasi peneliti menggunakan *use case diagram* yang bekerja dengan cara mendekripsikan tipe interaksi user dengan sebuah system melalui diagram. *Use case diagram* merupakan sebuah pemodelan untuk menggambarkan perilaku (tingkah laku) system aplikasi yang dibuat. Sebuah *use case* menggambarkan sebuah interaksi antara pengguna (*user*) dengan system yang ada. Berikut ini merupakan fungsi yang diterapkan pada *use case diagram* system aplikasi android dalam mengenkripsi dan dekripsi data file txt menggunakan algoritma RSA dan RC4.

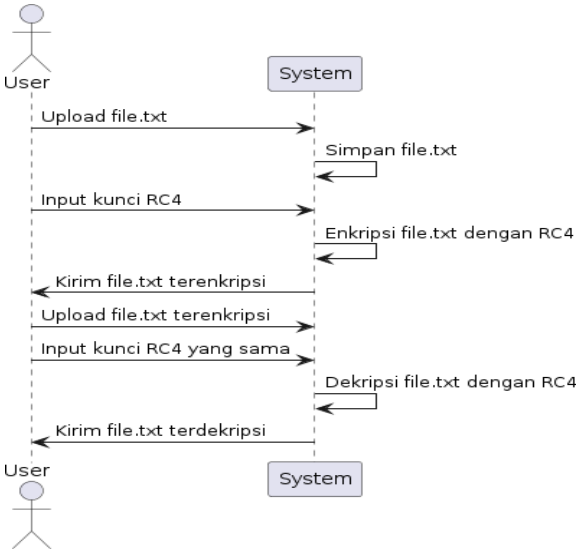
Gambar 2. *Use diagram*



E-ISSN : 2827-9425
P-ISSN : 2828-4666



Gambar 3. *Sequance Diagram* RSA



Gambar 4. *Sequance Diagram* RC4

4.2 Tampilan Halaman Aplikasi

Tampilan Menu Utama

Gambar 5. Tampilan Menu Utama

E-ISSN : 2827-9425
P-ISSN : 2828-4666

Halaman ini akan muncul saat program baru pertama kali dibuka, pada halaman ini hanya terdapat penjelasan tentang kriptografi.

Tampilan Menu RSA

Gambar 6. Tampilan Menu RSA

Pada tampilan halaman RSA, pengguna akan menguji kinerja enkripsi dan dekripsi algoritma RSA dari file txt untuk mengukur waktu eksekusi, penggunaan memori, ukuran file, banyak kata.

Tampilan Menu RC4

Gambar 7. Tampilan Menu RC4

Pada tampilan halaman RC4, pengguna akan menguji kinerja enkripsi dan dekripsi algoritma RC4 dari file txt untuk mengukur waktu eksekusi, penggunaan memori, ukuran file, dan banyak kata.

5. Kesimpulan

Berdasarkan dari hasil penelitian yang dilakukan peneliti dapat disimpulkan bahwa dapat diketahui RC4 cenderung lebih cepat dalam proses enkripsi dan dekripsi dibandingkan RSA. Hal ini disebabkan oleh perbedaan dalam kompleksitas algoritma dan metode enkripsi yang digunakan. Untuk keamanan RSA cenderung lebih aman dibandingkan RC4 karena RC4 telah terkena serangan kriptanalisis yang serius. RSA menggunakan kunci publik dan kunci pribadi , sementara RC4 hanya menggunakan kunci simetris. Penggunaan RSA sering digunakan untuk keperluan yang membutuhkan tingkat keamanan yang tinggi, seperti dalam pengamanan komunikasi melalui jaringan. RC4 meskipun cepat, telah ditinggalkan dalam

E-ISSN : 2827-9425
P-ISSN : 2828-4666

banyak aplikasi karena kelemahan keamanannya. Dengan demikian, pemilihan antara RSA dan RC4 tergantung pada kebutuhan spesifik pengguna, dengan pertimbangan utama antara keamanan dan kinerja.

DAFTAR PUSTAKA

[1] M. Rido Hasibuan, "Implementasi Algoritma Quicksort Untuk Pembangkitan Kunci Algoitma RSA Pada Pengamanan Data Audio," *J. Informatics, Electr. Electron. Eng.*, vol. 2, no. 1, pp. 18–25, 2022, doi: 10.47065/jieee.v2i1.392.

[2] Azlin, F. Musadat, and J. Nur, "Aplikasi Kriptografi Keamanan Data Menggunakan Algoritma Base64," J. Inform., vol. 7, no. 2, pp. 1–5, 2018.

[3] R. R. Rohmansyah and H. Nurwasito, "Pengembangan Aplikasi Mobile untuk Sistem Keamanan Kantor Menggunakan NFC (Near Field Communication) dan Wi-Fi (Studi Kasus : PT. Rahmi Ida Nusantara)," J. Pengemb. Teknlogi Inf. dan Ilmu Komput., vol. 2, no. 1, pp. 81–90, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/696

[4] N. S. Ulfah Indriani, Ommi Alfina, "Penerapan Algoritma Rsa Dan Affine Cipher Dalam Keamanan File Ms Word," vol. 01, no. 02, pp. 95–100, 2021, [Online]. Available: http://repository.potensi-utama.ac.id/jspui/handle/123456789/5074

[5] K. Fahmi, "RESOLUSI : Rekayasa Teknik Informatika dan Informasi Pengamanan Data Arsip Pada Balai Desa Sidodadi Menggunakan Kriptografi Modern RC4," Media Online, vol. 2, no. 2, pp. 58–66, 2021, [Online]. Available: https://djournals.com/resolusi

[6] F. Muhariza and N. Juliasari, "Implementasi Pengamanan Dokumen Menggunakan Kriptografi Dengan Algoritme Rivest Code 4 (Rc4) Berbasis Web Implementation of Document Security Using Cryptography With Web-Based Rivest Code 4 (Rc4)," vol. 2, no. September, pp. 131–139, 2023.

[7] S. B. H. Gmbh, "Definisi Perbandingan," pp. 1–23, 2016.

[8] M. S. Dairi, M. Setiani Asih, and corespondent author, "Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan Implementation Of RSA Cryptographic Algorithms in Library Information System Applications," Januari, vol. 2023, no. 2, pp. 214–223, 2022, [Online]. Available: https://jurnal.unity-academy.sch.id/index.php/jirsi/index%0Ahttp://creativecommons.org/licenses/by- sa/4.0/

[9] R. Maulana and R. M. Simanjorang, "Implementasi Kriptografi Pengamanan Data Pribadi Siswa SMA Swasta Jaya Krama Beringin Dengan Algoritma RC4," J. Nas. Komputasi dan Teknol. Inf., vol. 4, no. 6,

[10] pp. 377–383, 2021, doi: 10.32672/jnkti.v4i6.3533.