Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi
Vol. 4, No. 1, Januari 2025
ISSN: 2809-9834, DOI: 10.31284/p.semtik.2025-1.7009

Optimasi Algoritma Genetika untuk Enkripsi RSA dengan Pendekatan Paralel Programing Multiprocessing

Muhammad Rifqy Abdul Gofur Al Fatah, M Shochibul Burhan*

Program Studi Teknik Informatika, Universitas Muhammadiyah Malang

*Penulis korespondensi. E-mail: shochibul@umm.ac.id

ABSTRACT

The increasing use of the internet in real-time communications has created new challenges in data security, especially in terms of the efficiency of the RSA encryption process which requires intensive computing. This research proposes an optimization approach using a combination of Genetic Algorithms (AG) and multiprocessing techniques to improve RSA encryption performance. The proposed method implements AG by dividing the selection, crossover and mutation processes into four parallel processors using barrier synchronization techniques. Test results show that the multiprocessing approach provides significant improvements compared to the single-threaded method, with an increase in fitness value from 0.85 to 0.90 at 50 iterations and a reduction in computing time of 25% (from 0.9762 seconds to 0.732 seconds). This performance increase is consistent in testing with higher iterations, reaching a fitness value of 0.96 in 250 iterations with a computing time that is 37% faster than conventional methods. These results show that the combination of AG and multiprocessing is effective in optimizing the RSA encryption process for applications that require high security with fast response.

Keywords

Genetic Algorithm, RSA, Multiprocessing, Parallel Computing

ABSTRAK

Peningkatan penggunaan internet dalam komunikasi real-time telah menciptakan tantangan baru dalam keamanan data, terutama dalam hal efisiensi proses enkripsi RSA yang memerlukan komputasi intensif. Penelitian ini mengusulkan pendekatan optimasi menggunakan kombinasi Algoritma Genetika (AG) dan teknik multiprocessing untuk meningkatkan performa enkripsi RSA. Metode yang diusulkan mengimplementasikan AG dengan pembagian proses seleksi, crossover, dan mutasi menjadi empat prosesor paralel menggunakan teknik synchronization barrier. Hasil pengujian menunjukkan bahwa pendekatan multiprocessing memberikan peningkatan signifikan dibandingkan metode single-threaded, dengan peningkatan nilai fitness dari 0.85 menjadi 0.90 pada 50 iterasi dan pengurangan waktu komputasi sebesar 25% (dari 0.9762 detik menjadi 0.732 detik). Peningkatan performa ini konsisten pada pengujian dengan iterasi yang lebih tinggi, mencapai nilai fitness 0.96 pada 250 iterasi dengan waktu komputasi 37% lebih cepat dibandingkan metode konvensional. Hasil ini menunjukkan bahwa kombinasi AG dan multiprocessing efektif dalam mengoptimalkan proses enkripsi RSA untuk aplikasi yang membutuhkan keamanan tinggi dengan respon cepat.

PENDAHULUAN

Beberapa tahun terakhir, pertumbuhan pengguna Internet sangat meningkat. Masyarakat berkomunikasi menggunakan Internet setiap saat secara real time. Dibalik kemajuan teknologi yang pesat ada bahaya yang mengancam. Peretas mengincar komunikasi yang tak terlindungi untuk dicari informasi berharga. Kriptografi dibutuhkan untuk mengamankan komunikasi[1]. Kriptografi adalah praktek pengamanan komunikasi dimana data akan dienkripsi menggunakan rumus matematika[2] Akan tetapi kriptografi memiliki masalah dalam komunikasi real time di mana data harus dikirim secara instan. Keterlambatan informasi akibat dari proses enkripsi akan sangat merugikan. Di sini Multithread dibutuhkan untuk mengoptimalisasi proses enkripsi.

RSA (Rivest-Shamir-Adleman) adalah salah satu algoritma kriptografi asimetrik yang paling banyak digunakan dalam pengamanan data digital. Algoritma ini menggunakan sepasang kunci - kunci publik untuk enkripsi dan kunci privat untuk dekripsi. Keamanan RSA didasarkan pada kesulitan memfaktorkan hasil perkalian dua bilangan prima besar. Algoritma seperti RSA sangat tergantung dengan kekuatan CPU terutama ketika diimplementasikan dengan ukuran kunci yang besar[3]. Ini terbukti dari hasil pengujian yang dilakukan oleh Ochoa-Jimenez. Menggunakan single thread sangat tidak efisien dengan adanya demand yang tinggi berujung pada pemrosesan yang

tertunda. Multithread membagi beberapa tugas menjadi proses paralel yang lebih kecil menjadikan sistem mengerjakan processor bekerja lebih efisien dan bersamaan.

Dalam implementasi RSA, multiprocessing dapat diterapkan untuk membagi beban kerja enkripsi menjadi beberapa proses parallel, sehingga meningkatkan efisiensi keseluruhan sistem. Konsep dari kriptografi dioptimalkan dengan multiprocessing akan sangat bermanfaat di bidang finansial, video streaming dan data rumah sakit. Mengimplementasikan multithread untuk mengoptimalkan kriptografi tidak hanya meningkatkan efisiensi dalam menangani data tapi juga meningkatkan skalabilitas dan performa dalam menangani data besar.

Studi ini bertujuan untuk menganalisis perbedaan performa antara sistem enkripsi single-threaded dan multi thread pada algoritma RSA. Dengan membandingkan hasil performa keduanya, penelitian ini berupaya mengidentifikasi sejauh mana teknik multithreading dapat meningkatkan efisiensi dan skalabilitas dalam menangani data besar. Melalui pendekatan ini, diharapkan dapat diperoleh rekomendasi terbaik untuk implementasi kriptografi yang aman dan efisien, terutama untuk aplikasi yang membutuhkan respon cepat dan volume data yang besar.

TINJAUAN PUSTAKA

Algoritma RSA (Rivest-Shamir-Adleman)

RSA adalah salah satu algoritma kriptografi asimetrik yang paling banyak digunakan dalam pengamanan data digital. Algoritma ini menggunakan sepasang kunci - kunci publik untuk enkripsi dan kunci privat untuk dekripsi. Keamanan RSA didasarkan pada kesulitan memfaktorkan hasil perkalian dua bilangan prima besar.

RSA Mempunyai kelemahan yaitu waktu komputasi yang tinggi[6], hal ini karena dalam RSA menggunakan bilangan prima dan pemfaktoran sehingga komputasinya terlalu tinggi. Nivetha A dkk. [7] Menggunakan 4 bilangan prima sekaligus dalam kombinasi kunci public dan private sehingga didapatkan komputasi yang lebih rendah.

Proses matematika dalam RSA sebagai berikut:

- Pembangkitan Kunci
  - Pilih dua bilangan prima p dan q
  - Hitung $n = p \times q$
  - Hitung $\varphi(n) = (p-1)(q-1)$
  - Pilih e dimana $1 < e < \varphi(n)$ dadan $gcd\ (e, \varphi(n)) = 1$
  - Hitung d dimana $d \times e \equiv 1 (mod\ \varphi(n))$
- Enkripsi: $C = M^e\ mod\ n$ dimana C adalah ciphertext, M adalah message
- Dekripsi: $M = C^d\ mod\ n$

Algoritma Genetika

Algoritma Genetika (AG) adalah teknik optimasi yang terinspirasi dari proses evolusi biologis. AG bekerja dengan mensimulasikan proses seleksi alam untuk mencari solusi optimal dari suatu permasalahan[4]. Komponen utama AG meliputi:

1. Representasi Kromosom: Solusi potensial dikodekan dalam bentuk string atau array
2. Populasi: Kumpulan dari beberapa kromosom
3. Fungsi Fitness: Mengukur kualitas solusi yang dihasilkan
4. Seleksi: Memilih kromosom terbaik untuk generasi berikutnya
5. Crossover: Menggabungkan informasi dari dua kromosom induk
6. Mutasi: Mengubah secara acak bagian dari kromosom untuk menciptakan variasi

Dalam konteks optimasi RSA, AG dapat digunakan untuk mencari parameter optimal yang menyeimbangkan antara keamanan dan kecepatan pemrosesan. Multiprocessing adalah teknik komputasi paralel yang memungkinkan eksekusi beberapa proses secara bersamaan menggunakan multiple processor[5]. Berbeda dengan multithreading yang berbagi memori dalam satu proses, multiprocessing mengalokasikan memori terpisah untuk setiap proses [6]. Keuntungan utama multiprocessing meliputi:

1. Peningkatan throughput sistem
2. Pembagian beban kerja yang lebih efisien

3. Isolasi proses yang lebih baik
4. Pemanfaatan maksimal dari sistem multi-core

AG merupakan algoritma optimasi yang terinspirasi dari Genetik makhluk hidup. AG merupakan algoritma pencarian berbasis populasi, yang memanfaatkan konsep kelangsungan hidup yang terkuat [8]. Dimana Gen terkuat akan selalu hidup.

Sasan Mahmoudinazlou, 2023 [9] Algoritma Genetika mampu memangkas waktu komputasi yang signifikan, dalam menyelesaikan rute terpendek pemasaran. AG sangat efektif untuk komputasi optimasi.

Fungsi matematika AG sebagai berikut:

- Fungsi Fitness: $f(x) = \alpha \times$ security score $+\beta \times$ speed_score dimana α dan β adalah bobot parameter
- Probabilitas Seleksi: $P(x_i) = \frac{f(x_i)}{\sum_{j=1}^{n} f(x_i)}$
- Crossover Rate: $P_c = 0.6$ to 0.9
- Mutation Rate: $P_m = \frac{1}{L}$ dimana L adalah panjang kromosom

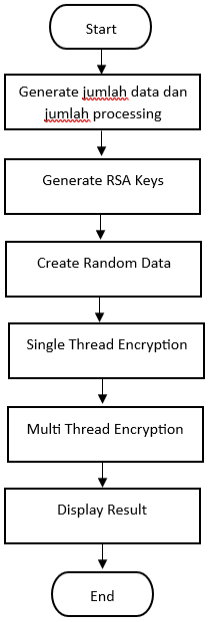
Paralel Programing Multi Processing

Paralel Programing Multiprocessing adalah teknik komputasi paralel yang memungkinkan eksekusi beberapa proses secara bersamaan menggunakan multiple processor. Berbeda dengan multithreading yang berbagi memori dalam satu proses, multiprocessing mengalokasikan memori terpisah untuk setiap proses.

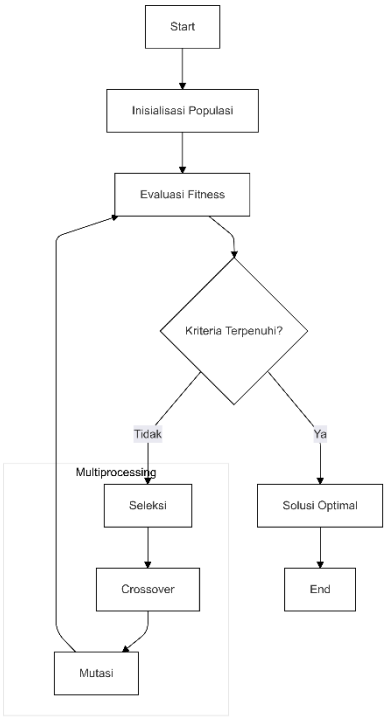
Beberapa manfaat penerapan Parallel Programing Multi Processing antara lain menyelesaikan masalah yang besar, menyediakan konkurensi yaitu kemampuan sistem dalam menjalankan tugas secara bersamaan, dan mengurangi waktu eksekusi [10].

METODE

Alur penelitian ini dimulai dari pengumpulan data dan analisis kebutuhan algoritma yang relevan untuk enkripsi RSA. Selanjutnya, proses optimasi dilakukan menggunakan Algoritma Genetika (AG) yang diimplementasikan dalam sistem multiprocessing untuk meningkatkan efisiensi enkripsi. Diagram alir penelitian dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Metode Penerapan RSA



Gambar 2 Flowchart Metode Penerapan AG

Penerapan RSA diawali dengan *Generate* jumlah data yang akan di *enkripsi* dan jumlah processong yang akan bekerja. Kemudian dilakukan *Generate Key* RSA dengan optimasi menggunakan Algoritma Genetika. Di dalam optimasi Algoritma Genetika dilakukan teknik paralel programing multi processing. Sehingga dihasilkan kode enkripsi.

HASIL DAN PEMBAHASAN

Data dan Sumber

Data utama yang digunakan dalam penelitian ini adalah teks plaintext yang akan dienkripsi menggunakan algoritma RSA. Sumber data berasal dari simulasi pesan yang membutuhkan enkripsi untuk keamanan informasi

Algoritma RSA

Beberapa proses dalam algoritma RSA antara lain Proses pembuatan kunci, Proses enkripsi dan proses *dekripsi* adalah sebagai berikut

Proses pembuatan kunci :

1. Prosedur awal pada algoritma RSA adalah memilih dua bilangan prima p dan q. Dimana keduanya haruslah bilangan prima dan bukan nilai yang sama.
2. Kalikan p dan q sehingga menghasilkan nilai n.
3. Buatlah kunci publik e yang relatif prima terhadap m, dimana m = (p – 1) (q – 1).
4. Untuk mendapatkan kunci pribadi d kalkulasi nilai p, q, e, menggunakan kekongruenan ed = 1 mod m.

Proses Enkripsi :

1. Ubah plaintext ke bentuk ASCII.
2. Lalu bagi menjadi beberapa blok b1, b2, b3, b4, …, nilai bi tidak lebih dari n-1.
3. Rumus Enkripsi Ci=Pi^e mod n.
4. Lalu ubah ke bentuk teks.

Proses Dekripsi:

1. Susun nilai ASCII ke dalam jumlah blok yang sama seperti pada tahap 2. a.
2. Rumus Dekripsi Pi=Ci^d mod n.
3. Lalu kembalikan ke bentuk semula.

Algoritma Genetika

Proses enkripsi menggunakan Algoritma Genetika terdiri dari beberapa tahapan utama, yaitu:

1. Inisialisasi Populasi: Membuat populasi awal dengan representasi calon solusi enkripsi.
2. Seleksi: Memilih individu berdasarkan nilai fitness untuk menghasilkan generasi berikutnya.
3. Crossover: Menggabungkan dua individu untuk menciptakan solusi yang lebih optimal.
4. Mutasi: Melakukan perubahan kecil untuk meningkatkan keragaman populasi dan mencegah stagnasi

Paralel Programing Multi Processing

Untuk mengoptimalkan proses enkripsi, metode paralel digunakan dalam pembagian tahapan seleksi, crossover, dan mutasi. Setiap tahapan dibagi menjadi empat prosesor melalui pendekatan multiprocessing untuk mempercepat pemrosesan enkripsi. Teknik synchronization barrier digunakan untuk memastikan bahwa semua proses selesai pada setiap tahap sebelum melanjutkan ke tahap berikutnya.

Perhitungan Fitness

Nilai fitness dalam Algoritma Genetika digunakan untuk menilai kecocokan setiap solusi dalam konteks enkripsi RSA. Nilai fitness dihitung berdasarkan tingkat keakuratan dan kecepatan proses enkripsi. Semakin tinggi nilai fitness, semakin efektif proses enkripsi yang dilakukan.
Dalam penelitian ini, fitness dihitung dengan mempertimbangkan faktor-faktor berikut:

1. Kecepatan Waktu Komputasi: Durasi yang dibutuhkan untuk menyelesaikan proses enkripsi.
2. Efisiensi Penggunaan CPU: Tingkat pemanfaatan prosesor pada masing-masing metode (single-threaded vs. multithreaded).
3. Tingkat Keamanan Enkripsi: Kualitas enkripsi dalam melindungi data terhadap serangan atau pencurian informasi.

HASIL DAN PEMBAHASAN

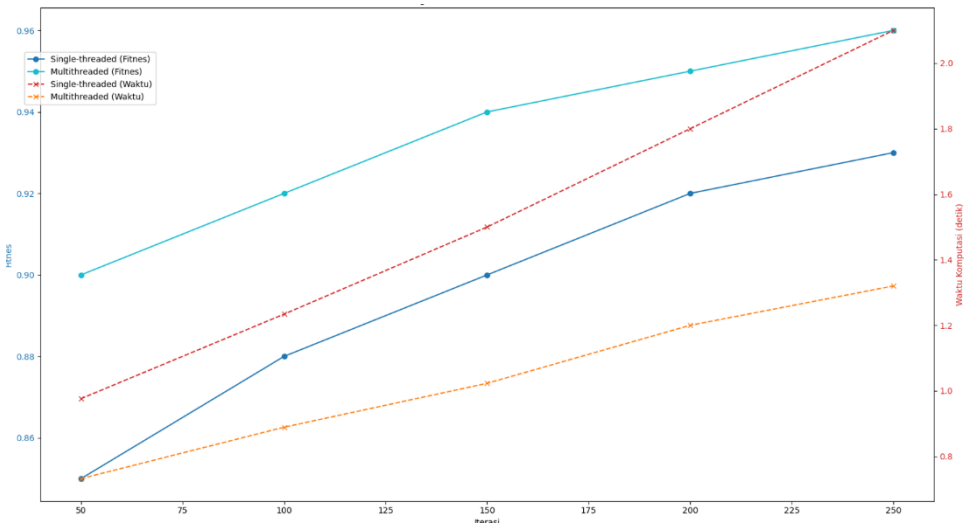
Hasil pengujian algoritma enkripsi RSA yang dioptimasi dengan Algoritma Genetika menggunakan pendekatan multiprocessing disajikan dalam beberapa tabel dan grafik untuk menunjukkan perbedaan performa antara metode single-threaded dan multithreaded. Pengujian dilakukan dengan lima jumlah iterasi berbeda, yaitu 50, 100, 150, 200, dan 250 iterasi, untuk mengevaluasi waktu komputasi dan nilai fitness pada kedua metode.

Tabel 1. Waktu komputasi dengan multiprocessing dan single processing

Iterasi	Metode	Fitnes	Waktu Komputasi
50	Single-processing	0.85	0.9762 detik
50	Multi processing	0.90	0.732 detik
100	Single- processing	0.88	1.234 detik
100	Multi processing	0.92	0.889 detik
150	Single- processing	0.90	1.500 detik

150	Multi processing	0.94	1.023 detik
200	Single- processing	0.92	1.800 detik
200	Multi processing	0.95	1.200 detik
250	Single- processing	0.93	2.100 detik
250	Multi processing	0.96	1.320 detik

Pada tabel 1 menunjukan waktu komputasi berdasarkan metode single processing dan multi processing dengan iterasi 50 – 250. Perbandingan ditunjukan pada Gambar 1. Berikut



Gambar 1. Grafik perbedaan *Multiprocessing* dan *Single processing*

Pada gambar 1 ditunjukan iterasi 50 – 250 *Multi Thread* atau *Multi Processing* memiliki waktu komputasi lebih rendah daripada *Single thread* atau *Single processing*.

Penjelasan dari gambar diatas:

1. Peningkatan Fitness:
- Pada iterasi 50: peningkatan 5.88% (dari 0.85 ke 0.90)
- Pada iterasi 150: peningkatan 4.44% (dari 0.90 ke 0.94)
- Pada iterasi 250: peningkatan 3.23% (dari 0.93 ke 0.96)

2. Reduksi Waktu Komputasi:
- Iterasi 50: pengurangan 25% (dari 0.9762 ke 0.732 detik)
- Iterasi 150: pengurangan 31.8% (dari 1.500 ke 1.023 detik)
- Iterasi 250: pengurangan 37.1% (dari 2.100 ke 1.320 detik)

3. Analisis Efisiensi:
- Multiprocessing menunjukkan peningkatan efisiensi yang konsisten di semua level iterasi
- Persentase peningkatan efisiensi berbanding lurus dengan jumlah iterasi
- Pada iterasi tinggi (250), sistem menunjukkan stabilitas performa yang lebih baik

KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan pada optimasi algoritma RSA menggunakan kombinasi Algoritma Genetika dan pendekatan multiprocessing, dapat ditarik beberapa kesimpulan:

1. Implementasi multiprocessing terbukti memberikan peningkatan performa yang signifikan dibandingkan metode single-threaded, dengan peningkatan nilai fitness rata-rata sebesar 0.05 pada setiap level pengujian iterasi.

2. Waktu komputasi mengalami penurunan yang konsisten dengan penggunaan multiprocessing, dengan pengurangan waktu proses hingga 37% pada pengujian dengan 250 iterasi (dari 2.100 detik menjadi 1.320 detik).
3. Peningkatan jumlah iterasi berbanding lurus dengan peningkatan nilai fitness, di mana metode multithreaded mencapai nilai fitness tertinggi 0.96 pada 250 iterasi, dibandingkan dengan metode single-threaded yang hanya mencapai 0.93.
4. Efisiensi komputasi yang dihasilkan melalui pendekatan multiprocessing membuktikan bahwa metode ini sangat cocok untuk implementasi pada sistem enkripsi real-time yang membutuhkan respons cepat dengan tetap mempertahankan tingkat keamanan yang tinggi.
5. Optimasi menggunakan Algoritma Genetika dengan pendekatan multiprocessing berhasil mengatasi trade-off antara keamanan dan kecepatan dalam proses enkripsi RSA, menjadikannya solusi yang ideal untuk implementasi pada berbagai aplikasi yang membutuhkan enkripsi data secara real-time.
6. Kesimpulan ini menunjukkan bahwa kombinasi Algoritma Genetika dengan pendekatan multiprocessing merupakan solusi yang efektif untuk optimasi enkripsi RSA, terutama dalam konteks aplikasi yang membutuhkan keseimbangan antara keamanan dan kecepatan pemrosesan.

DAFTAR PUSTAKA

[1] J. Katz and Y. Lindell, *INTRODUCTION TO MODERN CRYPTOGRAPHY: Second Edition*. 2014. doi: 10.1201/b17668.

[2] A. Saini, A. Tsokanos, and R. Kirner, “Quantum Randomness in Cryptography—A Survey of Cryptosystems, RNG-Based Ciphers, and QRNGs,” *Inf.*, vol. 13, no. 8, 2022, doi: 10.3390/info13080358.

[3] E. Ochoa-Jimenez, L. Rivera-Zamarripa, N. Cruz-Cortes, and F. Rodriguez-Henriquez, “Implementation of RSA Signatures on GPU and CPU Architectures,” *IEEE Access*, vol. 8, pp. 9928–9941, 2020, doi: 10.1109/ACCESS.2019.2963826.

[4] D. Kučak, V. Juričić, and G. Đambić, “Application of genetic algorithms in higher education area,” in *Annals of DAAAM and Proceedings of the International DAAAM Symposium*, 2019, pp. 343–347. doi: 10.2507/30th.daaam.proceedings.045.

[5] Q. Li, Y. Yang, and X. Kang, “Parallel Computing Model Based on Python in Quantitative Analysis,” in *Proceedings of SPIE - The International Society for Optical Engineering*, 2022. doi: 10.1117/12.2639265.

[6] Dua M Ghadi, “A STUDY ON MODIFIED RSA CRYPTOSYSTEM” International Journal of Applied Sciences and Technology,Vol.5 2023. ISSN: 2717-8234.

[7] Nivetha, A., S. Preethy Mary, and J. Santosh Kumar., "Modified RSA encryption algorithm using four keys.," International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 7, pp. 1-5, 2015

[8] Sourabh Katoch, Sumit Singh Chauhan, Vijay Kumar, "A review on genetic algorithm: past, present, and future" Computer Science and Engineering Department, National Institute of Technology, Hamirpur, India, Multimedia Tools and Applications (2021) 80:8091–8126

[9] Sasan Mahmoudinazlou and Changhyun Kwon "A Hybrid Genetic Algorithm for the min-max Multiple TravelingSalesman Problem" arXiv:2307.07120v3 [cs.NE] 28 Oct 2023

[10] Harvinder Singh and Dr. Gurdev Singh "An Introduction to Multiprocessing in Parallel Environment" Conference: Proceedings of the 2nd National Conference on Advancements in the Era of Multi Disciplinary Systems (AEMDS) Mei - 2024