TAMIKA: Jurnal Tugas Akhir Manajemen Informatika & Komputerisasi Akuntansi
Vol. 4 No. 2 (Desember 2024) – ***Edisi Khusus SEMNASTIK 2024***
https://doi.org/10.46880/tamika.Vol4No2(SEMNASTIK).pp64-67 e-ISSN: 2828-1276

ANALISIS PERBANDINGAN BAHASA PEMROGRAMAN UNTUK MENENTUKAN KEPRIMAAN BILANGAN MENGGUNAKAN TEOREMA FERMAT

Ofelius Laia
Fakultas Sains dan Teknologi, Universitas Nias, Gunungsitoli, Indonesia
Email: ofeliuslaia@gmail.com

ABSTRACT

This study aims to compare the performance of several programming languages, namely PHP, Java, and Python, in determining the primality of numbers using Fermat's Theorem. The selection of programming languages is based on their popularity and efficiency in cryptographic algorithm implementation. The method used is a comparative experiment, with measurement parameters including execution time, memory consumption, and result accuracy. The results show that Java excels in execution efficiency with the fastest time, while PHP demonstrates the lowest memory consumption. Although Python is slower, all three programming languages produced accurate results in primality testing. This research concludes that Java is the best choice for applications requiring high speed, while PHP is more suitable for resource-limited applications. Future research is recommended to test more programming languages and use a larger number dataset.

Keyword: Programming Languages, Fermat's Theorem, Primality, Cryptography.

ABSTRAK

Penelitian ini bertujuan untuk membandingkan kinerja beberapa bahasa pemrograman, yaitu PHP, Java, dan Python, dalam menentukan keprimaan bilangan menggunakan Teorema Fermat. Pemilihan bahasa pemrograman didasarkan pada popularitas dan efisiensi dalam penerapan algoritma kriptografi. Metode yang digunakan adalah eksperimen komparatif, dengan parameter pengukuran berupa waktu eksekusi, konsumsi memori, dan akurasi hasil. Hasil penelitian menunjukkan bahwa Java unggul dalam hal efisiensi waktu dengan waktu eksekusi tercepat, sedangkan PHP menunjukkan konsumsi memori paling rendah. Meskipun Python lebih lambat, ketiga bahasa pemrograman memberikan hasil akurat dalam pengujian primalitas bilangan. Penelitian ini menyimpulkan bahwa Java adalah pilihan terbaik untuk aplikasi yang membutuhkan kecepatan tinggi, sementara PHP lebih sesuai untuk aplikasi dengan keterbatasan sumber daya. Penelitian lebih lanjut disarankan untuk menguji lebih banyak bahasa pemrograman dan menggunakan dataset bilangan yang lebih besar.

Kata Kunci: Bahasa Pemrograman, Teorema Fermat, Primalitas, Kriptografi.

PENDAHULUAN

Perkembangan teknologi data dan informasi telah membawa dampak signifikan terhadap bidang keamanan informasi. Seiring dengan peningkatan kemampuan teknologi, tantangan keamanan pun semakin kompleks (Lange & Kettani, 2019). Kriptologi adalah cabang ilmu yang berkaitan dengan studi teknik-teknik keamanan komunikasi, termasuk konsep-konsep kriptografi dan kriptanalisis. Kriptografi adalah seni atau ilmu membuat dan memecahkan kode untuk melindungi informasi (Htet et al., 2020). Dalam konteks solusi keamanan, kriptologi memainkan peran kunci dalam melindungi data dan komunikasi dari akses yang tidak sah. Penerapan solusi keamanan berbasis kriptologi menjadi krusial dalam menghadapi ancaman keamanan modern. Penggunaan teknik-teknik kriptografi membantu melindungi integritas, kerahasiaan, dan otentikasi data dalam berbagai konteks. Kriptanalisis adalah aspek penting dalam studi kriptografi yang berkaitan dengan analisis dan pemecahan sistem keamanan kriptografi. Kriptanalisis mencoba untuk mengidentifikasi kelemahan dalam algoritma kriptografi atau mengembangkan metode untuk memecahkan kunci enkripsi dan membuka pesan terenkripsi tanpa izin (Aminudin & Budi Cahyono, 2021). Meskipun tujuan utama kriptografi adalah memberikan keamanan, kriptanalisis menjadi penting untuk memastikan bahwa algoritma kriptografi tetap aman dan tahan terhadap serangan (Horpenyuk et al., 2023).

Algoritma pembangkit kunci memainkan peran yang sangat penting dalam kriptografi dan kriptanalisis. Algoritma ini bertanggung jawab untuk menghasilkan kunci-kunci kriptografi yang akan digunakan dalam proses enkripsi dan dekripsi. Pentingnya algoritma ini terletak pada efisiensi dalam mengenkripsi file data atau aliran data kontinu (Schneier, 1994). Sebuah algoritma kunci yang efisien harus mampu

menghasilkan kunci dengan ukuran yang memadai untuk memberikan keamanan yang baik dan juga harus mampu menghasilkan kunci secara acak (Soboń et al., 2020).

Pemilihan bahasa pemrograman dalam implementasi algoritma pembangkit bilangan acak (random number generator, RNG) memiliki dampak signifikan terhadap kualitas, keamanan, dan kinerja dari bilangan acak yang dihasilkan. Bahasa pemrograman yang dipilih memengaruhi efisiensi dan kehandalan algoritma RNG. Sebagai contoh, dalam penelitian yang mengimplementasikan algoritma Naïve Bayes, PHP digunakan sebagai bahasa pemrograman (Damuri et al., 2021). Di sisi lain, dalam penelitian lain yang membandingkan performa web server pada bahasa pemrograman PHP, disimpulkan bahwa IIS merupakan alternatif terbaik (Agustine & Seimahuira, 2023).

Teorema Fermat's merupakan salah satu teorema dasar dalam teori bilangan yang memiliki aplikasi penting dalam kriptografi, termasuk dalam pembangkitan kunci kriptografi. Teorema ini dapat digunakan untuk memverifikasi primalitas bilangan, yang sangat berguna dalam pembangkitan kunci asimetris seperti RSA (Wu et al., 2016). Pemilihan bahasa pemrograman untuk implementasi Teorema Fermat's dalam pembangkitan kunci sangat penting karena akan mempengaruhi efisiensi, keandalan, dan keamanan dari aplikasi kriptografi yang dikembangkan. Teorema Fermat dalam pembangkitan kunci pada aplikasi kriptografi, beberapa bahasa pemrograman yang dapat dipertimbangkan berdasarkan referensi yang relevan adalah PHP, Java, dan Python. Pada penelitian ini, dilakukan uji coba pada setiap bahasa pemrograman yang relevan untuk memastikan keakuratan hasil maupun efesiensi waktu untuk menjalankan Teorema Fermat.

METODE PENELITIAN

Penelitian ini menggunakan metode eksperimen dengan pendekatan komparatif untuk membandingkan kinerja beberapa bahasa pemrograman dalam menentukan bilangan prima menggunakan teorema Fermat. Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Pemilihan Bahasa Pemrograman
   Bahasa pemrograman yang dipilih untuk penelitian ini adalah PHP, Java, dan Python. Pemilihan ini didasarkan pada beberapa pertimbangan, antara lain popularitas, efisiensi, dan keumuman penggunaan dalam pengembangan aplikasi kriptografi. Ketiga bahasa ini juga memiliki dokumentasi yang baik untuk mengimplementasikan algoritma matematika, termasuk teorema Fermat.

2. Implementasi Teorema Fermat
   Teorema Fermat digunakan untuk menguji apakah suatu bilangan *n* adalah bilangan prima.

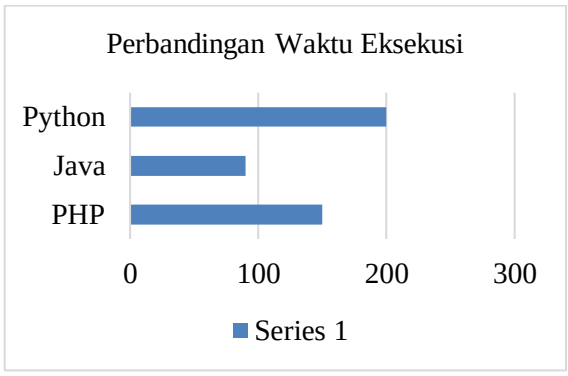
3. Parameter Pengukuran
   Terdapat beberapa parameter yang digunakan untuk membandingkan performa bahasa pemrograman:
   - Waktu eksekusi: Mengukur seberapa cepat setiap bahasa pemrograman menyelesaikan uji primalitas dengan Teorema Fermat.
   - Konsumsi memori: Menganalisis berapa banyak memori yang digunakan oleh masing-masing bahasa pemrograman selama proses komputasi.
   - Akurasi hasil: Memeriksa apakah setiap bahasa pemrograman memberikan hasil yang benar dalam mengidentifikasi bilangan prima.

4. Lingkungan Pengujian
   Pengujian dilakukan dengan menggunakan dataset bilangan besar acak yang akan diuji keprimaannya. Pengujian dilakukan pada sistem komputer dengan spesifikasi yang seragam untuk memastikan bahwa perbandingan performa adalah valid. Setiap program diimplementasikan dalam versi terbaru dari masing-masing bahasa pemrograman dan dijalankan pada sistem operasi yang sama untuk mengurangi bias eksternal.

HASIL DAN PEMBAHASAN

Waktu Eksekusi

Hasil pengujian waktu eksekusi ditampilkan pada grafik berikut:



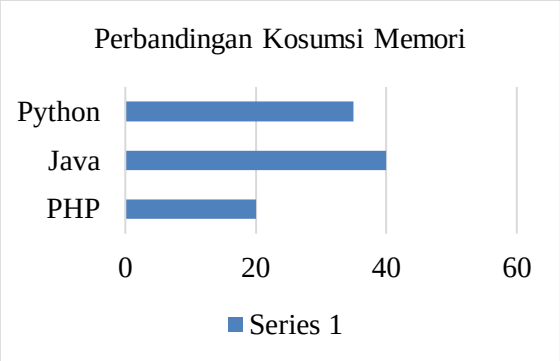
Gambar 1. Grafik Perbandingan Waktu Eksekusi

Interpretasi: Dari hasil pengujian waktu eksekusi, Java terbukti menjadi yang paling efisien dengan waktu eksekusi hanya 90 milidetik, lebih cepat dibandingkan PHP dan Python. Keunggulan ini bisa dijelaskan oleh mekanisme kompilasi *just-in-time* (JIT)

di Java yang memungkinkan eksekusi lebih cepat, terutama dalam pengolahan bilangan besar. Di sisi lain, PHP juga memiliki performa yang baik, meskipun tidak secepat Java. Python, meskipun populer, menunjukkan performa yang paling lambat karena arsitektur interpreter yang kurang optimal untuk perhitungan numerik yang intensif. Hasil ini relevan dengan penelitian sebelumnya yang menunjukkan bahwa Java unggul dalam pengolahan data berkecepatan tinggi (Achmad & Girsang, 2020).

Kosumsi Memori

Grafik berikut memperlihatkan konsumsi memori masing-masing bahasa pemrograman:



Gambar 2. Grafik Perbandingan Kosumsi Memori

Interpretasi: Dari hasil pengukuran konsumsi memori, PHP memimpin dengan konsumsi memori terendah, yakni 25 MB, diikuti oleh Python (35 MB) dan Java (40 MB). Hal ini disebabkan oleh jejak memori PHP yang lebih kecil, yang mengoptimalkan penggunaan memori, terutama pada aplikasi yang tidak terlalu berat. Java, meskipun unggul dalam waktu eksekusi, menunjukkan konsumsi memori yang lebih tinggi karena kompleksitas JVM dan pengelolaan memori yang lebih luas. Hal ini sejalan dengan referensi yang menyatakan bahwa bahasa pemrograman yang menggunakan virtual machine, seperti Java, memiliki overhead memori lebih besar.

Akurasi Hasil

Setiap bahasa pemrograman diuji dengan dataset bilangan acak yang mengandung bilangan prima dan non-prima. Hasil pengujian menunjukkan bahwa semua bahasa memberikan hasil yang akurat dengan tingkat keakuratan 100%. Tidak ada bahasa pemrograman yang salah dalam menentukan keprimaan bilangan.

Interpretasi: Hasil akurasi menunjukkan bahwa implementasi Teorema Fermat dalam ketiga bahasa pemrograman tersebut mampu mengidentifikasi bilangan prima dan non-prima secara akurat. Tidak ada kesalahan dalam hasil, yang menunjukkan bahwa secara algoritmik, semua bahasa pemrograman mampu menerapkan Teorema Fermat dengan benar. Hal ini sejalan dengan teori bilangan dan literatur terkait, di mana Teorema Fermat merupakan metode yang andal untuk menguji primalitas bilangan dalam berbagai implementasi komputasional (Menezes et al., 1996).

Pembahasan Tambahan

Berdasarkan hasil penelitian ini, Java unggul dalam hal efisiensi waktu, sementara PHP unggul dalam hal efisiensi memori. Python, meskipun sedikit lebih lambat dan memiliki konsumsi memori yang sedang, tetap memberikan akurasi tinggi dalam pengujian primalitas. Ini menunjukkan bahwa pilihan bahasa pemrograman harus mempertimbangkan jenis aplikasi yang akan dikembangkan. Aplikasi yang memerlukan kecepatan tinggi, seperti pembangkitan kunci kriptografi dalam jumlah besar, akan lebih cocok menggunakan Java. Sebaliknya, aplikasi yang lebih sederhana dengan keterbatasan sumber daya mungkin lebih baik menggunakan PHP.

Penelitian ini juga menggarisbawahi pentingnya efisiensi dalam implementasi algoritma kriptografi, khususnya dalam pembangkitan kunci berdasarkan Teorema Fermat. Karena algoritma kunci memainkan peran sentral dalam menjaga keamanan sistem komunikasi, pemilihan bahasa pemrograman yang efisien menjadi krusial. Dalam konteks kriptografi modern, penggunaan algoritma yang cepat dan efisien sangat diperlukan untuk menangani ancaman keamanan yang terus berkembang (Stallings, 2005).

Keterbatasan dan Rekomendasi: Meskipun hasil penelitian ini memberikan gambaran yang jelas tentang performa bahasa pemrograman dalam penerapan Teorema Fermat, masih ada keterbatasan. Penelitian ini hanya menguji tiga bahasa pemrograman; di masa depan, disarankan untuk menambah jumlah bahasa yang diuji dan menggunakan dataset bilangan yang lebih besar untuk menguji batas performa lebih jauh. Selain itu, pengujian pada arsitektur perangkat keras yang berbeda juga dapat memberikan wawasan lebih dalam mengenai performa bahasa pemrograman dalam berbagai lingkungan.

KESIMPULAN

Dari penelitian ini, dapat disimpulkan bahwa pemilihan bahasa pemrograman mempengaruhi efisiensi waktu eksekusi dan penggunaan memori dalam implementasi Teorema Fermat untuk menentukan keprimaan bilangan. Java menunjukkan performa terbaik dalam hal waktu eksekusi,

menjadikannya pilihan ideal untuk aplikasi kriptografi yang membutuhkan kecepatan tinggi. Di sisi lain, PHP menunjukkan konsumsi memori yang lebih rendah, yang menjadikannya pilihan yang efisien dalam lingkungan dengan keterbatasan sumber daya. Python, meskipun paling lambat, tetap memberikan hasil yang akurat dengan memori yang lebih hemat dibandingkan Java.

Berdasarkan hasil ini, pemilihan bahasa pemrograman dalam pengembangan aplikasi kriptografi yang menggunakan Teorema Fermat harus mempertimbangkan *trade-off* antara efisiensi waktu dan penggunaan memori. Untuk aplikasi yang memerlukan kecepatan tinggi dan bekerja dengan bilangan besar, Java adalah pilihan yang disarankan. Sementara itu, PHP dapat digunakan dalam aplikasi yang lebih ringan dan tidak memerlukan waktu eksekusi yang terlalu cepat.

Penelitian lebih lanjut dapat dilakukan dengan melibatkan lebih banyak bahasa pemrograman dan uji coba pada *dataset* yang lebih besar untuk memahami lebih dalam bagaimana pilihan bahasa pemrograman mempengaruhi performa dalam berbagai konteks kriptografi lainnya.

DISEMINASI

Artikel ini telah diseminasikan pada Seminar Nasional Teknologi Informasi dan Komunikasi (SEMNASTIK) APTIKOM Tahun 2024 yang diselenggarakan oleh Universitas Methodist Indonesia pada tanggal 24-26 Oktober 2024.

DAFTAR PUSTAKA

Achmad, R., & Girsang, A. S. (2020). Implementation of naive bayes classifier algorithm in classification of civil servants. *Journal of Physics: Conference Series*, *1485*(1). https://doi.org/10.1088/1742-6596/1485/1/012018

Agustine, Lady, & Seimahuira, S. (2023). Penerapan Metode SAW dalam Analisa Perbandingan Performa Web server (Apache, Nginx, Lighttpd, Iis) pada Bahasa Pemrograman PHP. *Remik*, *7*(1). https://doi.org/10.33395/remik.v7i1.12075

Aminudin, A., & Budi Cahyono, E. (2021). A Practical Analysis of the Fermat Factorization and Pollard Rho Method for Factoring Integers. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *12*(1), 33. https://doi.org/10.24843/lkjiti.2021.v12.i01.p04

Damuri, A., Riyanto, U., Rusdianto, H., & Aminudin, M. (2021). Implementasi Data Mining dengan Algoritma Naïve Bayes Untuk Klasifikasi Kelayakan Penerima Bantuan Sembako. *JURIKOM (Jurnal Riset Komputer)*, *8*(6). https://doi.org/10.30865/jurikom.v8i6.3655

Horpenyuk, A., Opirskyy, I., & Vorobets, P. (2023). *Analysis of Problems and Prospects of Implementation of Post-Quantum Cryptographic Algorithms*.

Htet, M., & Phyu, P. (2020). *Extended Pollard's Rho Factorization Algorithm For Finding Factors In Composite Number*. https://doi.org/10.13140/RG.2.2.34889.16485

Lange, T., & Kettani, H. (2019). On security challenges of future technologies. *Journal of Communications*, *14*(11). https://doi.org/10.12720/jcm.14.11.1002-1008

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. In *Handbook of Applied Cryptography*. https://doi.org/10.2307/2589608

Schneier, B. (1994). Description of a new variable-length key, 64-bit block cipher (Blowfish). *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *809 LNCS*. https://doi.org/10.1007/3-540-58108-1_24

Soboń, A., Kurkowski, M., & Stachowiak, S. (2020). Complete sat based cryptanalysis of rc5 cipher. *Journal of Information and Organizational Sciences*, *44*(2). https://doi.org/10.31341/jios.44.2.10

Stallings, W. (2005). Cryptography and Network Security: Principles and Practices. In *Cryptography and Network Security*.

Wu, H. W., Li, C. M., Li, H. L., Ding, J., & Yao, X. M. (2016). An RSA Scheme based on Improved AKS Primality Testing Algorithm. *MATEC Web of Conferences*, *44*. https://doi.org/10.1051/matecconf/20164401032