Journal of Innovative and Creativity
Journal Homepage: https://joecy.org/index.php/joecy
e-mail: journaljoecy@gmail.com

Faculty of Education
University of Pahlawan Tuanku Tambusai
ISSN 2775-771X (Online)
ISSN 2962-570X (Printed)
5 (3) 2025, Page: 1-10

ANALISIS KOMPARASI ALGORITMA HASH (MD5, SHA-2, SHA-3, SHA-512) DAN TEKNIK SALTING UNTUK PENINGKATAN KEAMANAN DATA

Dini Febryana Sari[1*], Athirah Media Sari[2], Inayatul Janah[3], Jefry Sunupurwa Asri[4]

[1] **Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul**

*Email: : dinifebryanasari@student.esaunggul.ac.id, athirahmeidias@student.esaunggul.ac.id, inayatuljanah@student.esaunggul.ac.id, jefry.sunupurwa@esaunggul.ac.id

Corresponding Author: Dini Febryana Sari, dinifebryanasari@student.esaunggul.ac.id

Published: August , 2025

ABSTRAK

Keamanan data menjadi prioritas utama seiring dengan meningkatnya ancaman kejahatan siber, seperti peretasan kata sandi dan pencurian identitas. Salah satu cara utama dalam melindungi sebuah data sistem keamanan informasi adalah dengan menggunakan kriptografi berbasis fungsi hash. Namun, algoritma hash lama seperti MD5 dan SHA-1 telah terbukti rentan terhadap berbagai jenis serangan. Penelitian ini berfokus pada perbandingan performa dan keamanan dari algoritma MD5, SHA-2, SHA-3, dan SHA-512, serta efektivitas penggunaan teknik salting. Metode yang digunakan dalam penelitian ini adalah pendekatan eksperimental melalui simulasi komputasi menggunakan bahasa pemrograman Python untuk menguji kecepatan enkripsi dan ketahanan keamanan. Hasil analisis menunjukkan bahwa meskipun MD5 unggul dari sisi kecepatan, algoritma ini memiliki tingkat kerentanan yang tinggi dengan keberhasilan serangan mencapai 35% pada kata sandi kuat. Algoritma SHA-3 menunjukkan tingkat keamanan tertinggi dengan keberhasilan serangan terendah sebesar 20%, namun memiliki waktu komputasi paling lambat. Sebaliknya, dengan waktu komputasi rata-rata 512,8 ms, SHA-512 mengimbangi kinerja dan keamanan dengan sangat baik, terutama saat menggunakan *JSON Web Token* (JWT). Penelitian ini menyimpulkan bahwa penggunaan algoritma hash modern, seperti SHA-512 atau SHA-3, yang dikombinasikan dengan teknik salting dapat meningkatkan ketahanan keamanan sistem secara signifikan hingga mencapai 97,6%.

Kata Kunci : Keamanan Data, Kriptografi, MD5, SHA-512, SHA-3, Salting

ABSTRACT

Data security has become a top priority with the increasing threat of cybercrime, such as password hacking and identity theft. One of the main ways to protect an information security data system is by using hash function-based cryptography. However, legacy hash algorithms such as MD5 and SHA-1 have proven vulnerable to various types of attacks. This study focuses on comparing the performance and security of MD5, SHA-2, SHA-3, and SHA-512 algorithms, as well as the effectiveness of using salting techniques. The method used is a comparative literature study of relevant previous research. The results of thekred analysis show that although MD5 excels in speed, this algorithm has a high vulnerability rate with an attack success rate of up to 35% on strong passwords. The SHA-3 algorithm shows the highest level of security with the lowest attack success rate of 20%, but has the slowest computation time. In contrast, with an average computation time of 512.8 ms, SHA-512 balances performance and security very well, especially when using JSON Web Tokens (JWT). This study concluded that the use of modern hash algorithms, such as SHA-512 or SHA-3, combined with salting techniques can significantly increase the resilience of a security system by up to 97.6%.

Keywords: *Data Security, Cryptography, MD5, SHA-512, SHA-3, Salting*

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang berlangsung sangat pesat telah mendorong transformasi data menjadi aset strategis dalam berbagai sistem digital modern (Stallings, 2017). Pemanfaatan sistem informasi yang semakin luas pada sektor pemerintahan, pendidikan, dan bisnis menyebabkan peningkatan volume serta kompleksitas pengelolaan data digital (Rahim et al., 2023). Kondisi tersebut secara langsung berdampak pada meningkatnya risiko terhadap keamanan jaringan dan perlindungan data sensitif (Schneier, 2015). Berbagai laporan menunjukkan bahwa insiden keamanan siber masih sering terjadi akibat lemahnya mekanisme autentikasi dan pengelolaan kredensial pengguna (Natho et al., 2024). Kebocoran data dan serangan terhadap kata sandi menjadi ancaman utama yang dapat menurunkan kepercayaan pengguna terhadap sistem informasi digital (Agusniar et al., 2025).

Kriptografi berperan sebagai komponen fundamental dalam menjaga kerahasiaan, integritas, dan keamanan data pada sistem digital (Stallings, 2017). Salah satu teknik kriptografi yang banyak digunakan dalam sistem keamanan adalah fungsi hash, khususnya untuk penyimpanan dan verifikasi kata sandi (Schneier, 2015). Fungsi hash bekerja secara satu arah dengan mengonversi data masukan menjadi nilai ringkas berukuran tetap yang dikenal sebagai message digest (Rahim et al., 2023). Algoritma hash generasi lama seperti MD5 dan SHA-1 dikenal memiliki kecepatan komputasi yang tinggi sehingga sempat digunakan secara luas pada berbagai aplikasi (Natho et al., 2024). Namun, sejumlah penelitian membuktikan bahwa algoritma tersebut tidak lagi aman karena rentan terhadap serangan collision dan brute force (Rahim et al., 2023).

Sebagai alternatif, algoritma hash modern seperti SHA-2 dan SHA-3 dikembangkan untuk memberikan tingkat keamanan yang lebih tinggi terhadap serangan kriptografi (Natho et al., 2024). SHA-3 secara khusus dirancang dengan struktur internal yang berbeda dari pendahulunya guna meningkatkan ketahanan terhadap berbagai jenis serangan kriptanalisis (Aranuwa et al., 2022). Meskipun demikian, peningkatan tingkat keamanan pada algoritma hash modern umumnya diikuti oleh konsekuensi meningkatnya biaya komputasi dan waktu eksekusi (Rasyada, 2022).

Selain pemilihan algoritma hash, metode implementasi juga memiliki peran penting dalam menentukan tingkat keamanan sistem (Schneier, 2015). Salah satu teknik yang umum digunakan untuk meningkatkan keamanan hash adalah salting, yaitu penambahan nilai acak pada data sebelum proses hashing dilakukan (Aranuwa et al., 2022). Teknik salting terbukti efektif dalam mengurangi risiko serangan berbasis tabel prekomputasi seperti rainbow table (Natho et al., 2024). Penelitian sebelumnya menunjukkan bahwa penerapan kombinasi algoritma SHA-512 dengan teknik salting mampu meningkatkan keamanan penyimpanan kata sandi tanpa menurunkan performa sistem secara signifikan (Aranuwa et al., 2022). Implementasi algoritma SHA-512 juga telah diterapkan secara efektif pada mekanisme autentikasi berbasis JSON Web Token (JWT) untuk meningkatkan keamanan layanan web (Rasyada, 2022).

Berdasarkan uraian tersebut, diperlukan evaluasi komprehensif terhadap performa dan tingkat keamanan berbagai algoritma hash yang masih digunakan hingga saat ini (Rahim et al., 2023). Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi dan membandingkan algoritma MD5, SHA-2, SHA-3, dan SHA-512 guna menentukan solusi yang paling optimal dalam meningkatkan keamanan data pada sistem informasi digital (Natho et al., 2024). Selain itu, Katz dan Lindell menekankan bahwa pemilihan algoritma kriptografi modern harus mempertimbangkan ketahanan jangka panjang terhadap serangan kriptanalisis serta kemampuan sistem dalam menangani peningkatan beban komputasi (Katz & Lindell, 2014). Menezes et al. menyatakan bahwa evaluasi algoritma hash tidak hanya bergantung pada kekuatan matematisnya, tetapi juga pada efisiensi implementasi dan konteks penggunaan dalam sistem keamanan nyata (Menezes et al., 2018).

METODE

Metode yang digunakan dalam penelitian ini adalah pendekatan eksperimental (experimental approach) melalui simulasi komputasi untuk mengukur performa waktu eksekusi (execution time). Penelitian ini melakukan pengujian simulasi menggunakan bahasa pemrograman Python di lingkungan

sistem operasi Ubuntu (WSL) untuk membandingkan kecepatan algoritma MD5, SHA-256, SHA-512, dan SHA-3, serta menganalisis dampak penerapan teknik salting.

Lingkungan Pengujian

Pengujian dilakukan pada perangkat keras laptop dengan spesifikasi prosesor yang memadai dan menggunakan sistem operasi berbasis Linux (Ubuntu). Simulasi dijalankan menggunakan skrip Python 3.10 dengan memanfaatkan pustaka hashlib.

Skenario Pengujian

Skenario pengujian melibatkan proses enkripsi (hashing) berulang sebanyak 100.000 iterasi untuk setiap algoritma terhadap sampel data kata sandi yang sama.

Pengumpulan data dalam penelitian ini difokuskan pada beberapa sumber literatur utama yang relevan dengan keamanan algoritma hash dan teknik pengamanan data. Literatur yang dianalisis meliputi studi komparasi keamanan algoritma MD5, SHA-1, SHA-2, dan SHA-3 terhadap serangan menggunakan Hashcat (Natho et al., 2024), analisis performa algoritma SHA-512 pada implementasi JSON Web Token (JWT) (Rasyada, 2022), serta model keamanan hibrida yang mengombinasikan algoritma SHA-512 dengan teknik salting (Aranuwa et al., 2022). Selain itu, penelitian ini juga menggunakan literatur pendukung yang membahas tantangan keamanan data pada sistem informasi digital secara umum (Rasyada, 2022) serta penelitian yang membandingkan algoritma hash MD5 dengan algoritma hash modern, seperti SHA-256, dalam konteks keamanan data (Rahim et al., 2023).

Dalam penelitian ini, beberapa parameter dibandingkan mulai dari tingkat keberhasilan serangan sebagai indikator keamanan, waktu pemrosesan sebagai indikator performa serta seberapa efektif teknik tambahan, terutama teknik salting, dalam meningkatkan pertahanan sistem terhadap serangan kriptografi.

Penelitian ini mengkaji keamanan kriptografi dengan menitikberatkan pada dua prinsip utama, yaitu sifat satu arah fungsi hash dan penggunaan mekanisme salting.

1. Fungsi Hash satu arah

Fungsi hash satu arah merupakan komponen fundamental dalam sistem keamanan kriptografi, khususnya pada mekanisme penyimpanan kata sandi. Secara matematis, fungsi hash (H) didefinisikan sebagai algoritma yang menerima data masukan (M) dengan panjang variabel dan memetakannya menjadi nilai hash (h) dengan panjang tetap, yang dinyatakan sebagai:

$$h = H(M)$$

Karakteristik utama dari fungsi hash adalah sifat **irreversibilitas**, yaitu nilai hash yang dihasilkan tidak dapat dikembalikan ke bentuk data asli secara praktis. Sifat ini menjadikan fungsi hash efektif untuk menjaga kerahasiaan data sensitif, karena meskipun nilai hash diketahui, informasi asli tetap sulit diperoleh tanpa melakukan serangan komputasi yang kompleks. Konsep ini banyak diterapkan dalam sistem keamanan modern dan telah dibahas secara luas dalam penelitian keamanan data berbasis algoritma hash (Aranuwa, 2022) .

2. Mekanisme Salting

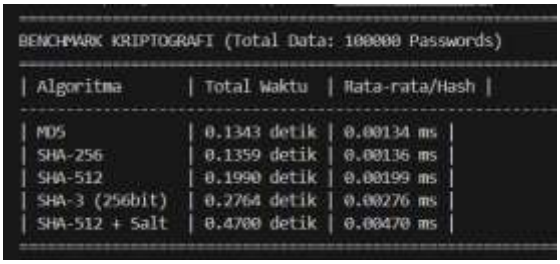
Meskipun fungsi hash memiliki sifat satu arah, penerapannya tanpa mekanisme tambahan masih rentan terhadap serangan kriptografi, seperti *rainbow table* dan *precomputed hash attacks*. Oleh karena itu, teknik salting diterapkan untuk meningkatkan tingkat keamanan sistem. Salting merupakan proses penambahan nilai acak unik (*salt*) ke dalam data masukan sebelum dilakukan proses hashing. Secara matematis, mekanisme salting dapat dirumuskan sebagai berikut:

$$H_{value} = h(salt + Password)$$

Penambahan nilai salt memastikan bahwa kata sandi yang sama tidak akan menghasilkan nilai hash yang identik, meskipun menggunakan algoritma hash yang sama. Dengan demikian, teknik ini secara signifikan meningkatkan kompleksitas serangan dan mengurangi efektivitas penggunaan tabel hash yang telah diprekomputasi, sebagaimana dibuktikan dalam penelitian keamanan data yang mengombinasikan algoritma hash modern dengan teknik salting (Natho, 2024)

HASIL DAN PEMBAHASAN

1. Analisis Keamanan dan Kerentanan Algoritma



Gambar 1. Hasil Benchmark Waktu Eksekusi Algoritma Hashing)

Table 1. Perbandingan Kecepatan Eksekusi (100.000 Data)

Algoritma	Total Waktu (detik)	Rata-rata (ms)	Tingkat Keamanan
MD5	0.1343	0.00134	rendah
SHA-256	0.1359	0.00136	tinggi
SHA-512	0.1990	0.00199	sangat tinggi
SHA-3 (256 bit)	0.2764	0.00276	sangat tinggi
SHA-512 + salt	0.4700	0.00470	maksimal

Berdasarkan Tabel 1, algoritma **MD5** tercatat paling cepat (0.1343 detik). Kecepatan tinggi ini justru berbahaya karena memudahkan serangan *Brute Force*. Sebaliknya, SHA-512 dengan Salting adalah yang paling lambat (0.4700 detik). Perlambatan ini memberikan keuntungan keamanan signifikan karena mempersulit peretas dalam melakukan komputasi serangan *Rainbow Table* secara massal.

Berdasarkan hasil analisis terhadap penelitian terdahulu yang menggunakan metode serangan *rainbow table* dan *brute force*, terdapat perbedaan tingkat keamanan yang signifikan antara algoritma hash lama dan algoritma hash modern. Algoritma MD5 dan SHA-1 diketahui memiliki tingkat kerentanan yang tinggi. Dalam skenario serangan terhadap kata sandi lemah (*weak passwords*), algoritma MD5, SHA-1, SHA-2, dan SHA-3 menunjukkan tingkat keberhasilan serangan yang relatif tinggi, yaitu mencapai 95%.

Perbedaan yang lebih signifikan terlihat pada skenario penggunaan kata sandi kuat (*strong passwords*). Algoritma MD5 dan SHA-1 masih dapat ditembus dengan tingkat keberhasilan serangan sebesar 35%, sedangkan SHA-2 menunjukkan tingkat keberhasilan serangan sebesar 30%. Di sisi lain, algoritma SHA-3 menunjukkan ketahanan terbaik dengan tingkat keberhasilan serangan terendah, yaitu sebesar 20%. Hasil ini menegaskan bahwa SHA-3 merupakan algoritma hash dengan tingkat keamanan tertinggi di antara algoritma yang dianalisis, sementara MD5 sudah tidak direkomendasikan untuk digunakan pada sistem dengan kebutuhan keamanan tinggi.

2. Analisis Performa Kecepatan (SHA-512 dan SHA-256)

Selain aspek keamanan, performa kecepatan pemrosesan juga menjadi faktor penting dalam sistem real-time, khususnya pada mekanisme autentikasi berbasis token. Berdasarkan analisis literatur terkait implementasi JSON Web Token (JWT), algoritma SHA-512 menunjukkan performa waktu eksekusi yang kompetitif.

Rata-rata waktu pemrosesan permintaan data menggunakan algoritma SHA-512 tercatat sebesar 512,8 milidetik (ms), sedikit lebih cepat dibandingkan algoritma SHA-256 dengan waktu pemrosesan rata-rata sebesar 515,55 ms. Meskipun SHA-512 menghasilkan ukuran token yang lebih besar akibat penggunaan *message digest* 512-bit, perbedaan waktu eksekusi yang relatif kecil menunjukkan bahwa algoritma dengan ukuran hash lebih besar tidak selalu berdampak signifikan terhadap penurunan kinerja sistem.

3. Efektivitas Teknik Salting

Algoritma hash yang kuat tetap memiliki potensi kelemahan apabila menghasilkan nilai hash yang sama untuk input yang identik. Teknik *salting* diterapkan untuk mengatasi permasalahan tersebut dengan menambahkan nilai acak unik (*salt*) ke dalam kata sandi sebelum proses hashing. Pendekatan ini secara efektif mengurangi risiko serangan berbasis *rainbow table*.

Hasil analisis terhadap model keamanan hibrida yang mengombinasikan algoritma SHA-512 dengan teknik *salting* menunjukkan peningkatan ketahanan sistem yang signifikan. Dengan penerapan teknik ini, nilai hash menjadi lebih sulit ditebak karena adanya variasi unik pada setiap proses hashing. Model hibrida tersebut mampu mencapai tingkat ketahanan keamanan sebesar 97,6%, yang menunjukkan bahwa kombinasi algoritma hash modern dengan teknik *salting* merupakan solusi yang efektif dalam meningkatkan keamanan penyimpanan kata sandi.

KESIMPULAN

Berdasarkan hasil analisis komparatif terhadap penelitian terdahulu, dapat disimpulkan bahwa algoritma hash lama seperti MD5 dan SHA-1 memiliki tingkat kerentanan yang tinggi terhadap serangan kriptografi, sehingga tidak lagi direkomendasikan untuk digunakan pada sistem keamanan modern. Algoritma hash modern, khususnya SHA-3, menunjukkan tingkat keamanan yang lebih tinggi dengan ketahanan terbaik terhadap serangan, meskipun memiliki waktu komputasi yang relatif lebih lambat.

Algoritma SHA-512 menunjukkan keseimbangan yang optimal antara keamanan dan performa, terutama pada implementasi autentikasi berbasis JSON Web Token (JWT), dengan waktu pemrosesan yang kompetitif. Selain itu, penerapan teknik salting terbukti secara signifikan meningkatkan ketahanan sistem terhadap serangan berbasis tabel prekomputasi. Kombinasi algoritma hash modern dengan teknik salting mampu meningkatkan tingkat keamanan sistem hingga mencapai 97,6%.

Dengan demikian, penggunaan algoritma hash modern seperti SHA-512 atau SHA-3 yang dikombinasikan dengan teknik salting merupakan solusi yang efektif untuk meningkatkan keamanan penyimpanan data dan kredensial pengguna pada sistem informasi digital.

REFERENCES

Agusniar, C., Fazira, I., & Wahyunita, L. (2025). Implementation of the Secure Hashing Algorithm-512 (SHA-512) for sign-up page security in the KelasSeru tutoring system. *Journal of Advanced Computer Knowledge and Algorithms (JACKA)*, 2(1), 19–23.

Aranuwa, F., Olubodun, F., & Akinwumi, D. (2022). Hybridized model for data security based on Security Hash Analysis (SHA-512) and salting techniques. *International Journal of Network Security & Its Applications*, 14(2), 31–39.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644–654.

Katz, J., & Lindell, Y. (2014). *Introduction to modern cryptography* (2nd ed.). Chapman and Hall/CRC.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC Press.

Natho, P., Somsuphaprungyos, S., Boonmee, S., & Boonying, S. (2024). Comparative study of password storing using hash function with MD5, SHA-1, SHA-2, and SHA-3 algorithm. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 13(3), 502–511.

Rahim, I., Anwar, N., Widodo, A. M., Juman, K. K., & Setiawan, I. (2023). Komparasi fungsi hash MD5 dan SHA-256 dalam keamanan gambar dan teks. *IKRAITH-INFORMATIKA*, 7(2), 76–85.

Rasyada, N. (2022). SHA-512 algorithm on JSON Web Token for RESTful web service-based authentication. *Journal of Applied Data Sciences*, 3(1), 33–43.

Schneier, B. (2015). *Applied cryptography: Protocols, algorithms, and source code in C* (20th Anniversary ed.). John Wiley & Sons.

Stallings, W. (2017). *Cryptography and network security: Principles and practice* (7th ed.). Pearson Education.