DOI: XXXXXXXXXX

ANALISIS PERBANDINGAN ALGORITMA HASH SHA-1 DAN SHA-256 MENGGUNAKAN BAHASA PEMROGRAMAN PHP

[1]Fidamayasari
[1]Sistem Informasi, Fakultas Teknik dan Ilmu Komputer, Universitas Islam Indragiri,
Email: [1]fidamayasari69@gmail.com

ABSTRAK

Penelitian ini membahas perbandingan algoritma hash SHA-1 dan SHA-256 yang diimplementasikan menggunakan bahasa pemrograman PHP. Tujuan dari penelitian ini adalah untuk menganalisis perbedaan kedua algoritma tersebut dari aspek keamanan, panjang output hash, serta performa komputasi. Metode yang digunakan dalam penelitian ini adalah studi literatur yang diperkuat dengan pengujian langsung (eksperimen) menggunakan fungsi bawaan PHP, yaitu sha1() dan hash('sha256', ...). Pengujian dilakukan dengan menggunakan data input yang sama dan diulang sebanyak 10 kali untuk setiap algoritma guna memperoleh hasil yang lebih konsisten. Hasil penelitian menunjukkan bahwa kedua algoritma menghasilkan nilai hash yang konsisten untuk setiap input yang sama, sehingga bersifat deterministik. Namun, SHA-256 memiliki panjang output yang lebih besar dibandingkan SHA-1, yaitu 256-bit berbanding 160-bit, sehingga memberikan tingkat keamanan yang lebih tinggi. Dari sisi performa, hasil pengujian menunjukkan adanya perbedaan waktu eksekusi, di mana SHA-256 memiliki waktu pemrosesan yang lebih kecil dibandingkan SHA-1, meskipun perbedaan tersebut tidak signifikan dalam penggunaan aplikasi skala umum. Berdasarkan hasil kajian literatur dan pengujian, SHA-1 telah dinyatakan tidak lagi aman digunakan karena rentan terhadap collision attack, sedangkan SHA-256 masih direkomendasikan dalam standar keamanan modern seperti SSL/TLS, digital signature, dan sistem blockchain. Dengan demikian, dapat disimpulkan bahwa SHA-256 lebih unggul dibandingkan SHA-1 dalam aspek keamanan dan lebih direkomendasikan untuk implementasi sistem informasi berbasis PHP.

Kata Kunci: SHA-1, SHA-256, PHP, Kriptografi, Hash Function

ABSTRACT

This study discusses the SHA-1 and SHA-256 hash algorithms implemented using the PHP programming language. The purpose of this study is to analyze the differences between the two algorithms in terms of security, hash output length, and computational performance. The method used in this study is a literature study reinforced by direct testing (experiments) using PHP's built-in functions, namely sha1() and hash('sha256', ...). Testing was carried out using the same input data and repeated 10 times for each algorithm to obtain more consistent results. The results show that both algorithms produce consistent hash values for each identical input, thus being deterministic. However, SHA-256 has a larger output length than SHA-1, namely 256-bit compared to 160-bit, thus providing a higher level of security. In terms of performance, the test results show a difference in execution time, where SHA-256 has a shorter processing time than SHA-1, although the difference is not significant in general-scale application use. Based on the results of literature reviews and testing, SHA-1 has been declared no longer safe to use because it is vulnerable to collision attacks, while SHA-256 is still recommended in modern security standards such as SSL/TLS, digital signatures, and blockchain systems. Thus, it can be concluded that SHA-256 is superior to SHA-1 in terms of security and is more recommended for implementing PHP-based information systems.

Keywords: SHA-1, *SHA-256, PHP, Cryptography, Hash Function.*

DOI: XXXXXXXXXX

1 PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat telah membawa dampak signifikan terhadap berbagai aspek kehidupan, termasuk dalam hal keamanan data. Di era digital saat ini, data menjadi aset yang sangat berharga bagi individu, organisasi, maupun institusi. Oleh karena itu, perlindungan terhadap data dari berbagai ancaman seperti pencurian, manipulasi, maupun akses tidak sah menjadi suatu kebutuhan yang tidak dapat diabaikan. Salah satu teknik yang широко digunakan dalam menjaga keamanan data adalah penggunaan algoritma kriptografi, khususnya fungsi hash. Fungsi hash merupakan suatu metode dalam kriptografi yang digunakan untuk mengubah data masukan (plaintext) menjadi nilai keluaran berupa rangkaian karakter dengan panjang tetap yang disebut sebagai hash value atau message digest. Fungsi hash yang baik memiliki beberapa karakteristik utama, seperti deterministik, cepat dalam proses komputasi, serta memiliki sifat one-way (tidak dapat dikembalikan ke bentuk semula). Selain itu, fungsi hash juga diharapkan memiliki tingkat ketahanan yang tinggi terhadap collision, yaitu kondisi di mana dua input berbeda menghasilkan nilai hash yang sama.

Dalam implementasinya, terdapat berbagai jenis algoritma hash yang telah dikembangkan, di antaranya adalah SHA-1 (Secure Hash Algorithm 1) dan SHA-256 (Secure Hash Algorithm 256-bit) yang merupakan bagian dari keluarga algoritma SHA (Secure Hash Algorithm). SHA-1 sebelumnya merupakan salah satu algoritma hash yang secara luas digunakan dalam berbagai aplikasi keamanan, seperti tanda tangan digital dan sertifikat SSL. Namun, seiring dengan perkembangan teknologi dan meningkatnya kemampuan komputasi, SHA-1 mulai dianggap tidak lagi aman karena telah ditemukan celah keamanan berupa collision attack yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab. Sebagai respons terhadap kelemahan tersebut, dikembangkan algoritma hash yang lebih kuat, salah satunya adalah SHA-256 yang merupakan bagian dari SHA-2. SHA-256 menghasilkan nilai hash dengan panjang 256-bit, yang secara signifikan lebih panjang dibandingkan SHA-1 yang hanya menghasilkan 160-bit. Panjang hash yang lebih besar ini memberikan tingkat keamanan yang lebih tinggi karena memperkecil kemungkinan terjadinya collision serta meningkatkan kompleksitas serangan brute force. Dalam konteks pengembangan perangkat lunak, khususnya menggunakan bahasa pemrograman PHP, kedua algoritma hash ini masih sering digunakan dalam berbagai implementasi, seperti pengamanan password, validasi data, serta integritas file. PHP menyediakan fungsi bawaan seperti sha1() dan hash('sha256', data) yang memudahkan pengembang dalam mengimplementasikan kedua algoritma tersebut. Namun demikian, pemilihan algoritma yang tepat tetap menjadi hal yang penting, mengingat perbedaan karakteristik, tingkat keamanan, serta performa yang dimiliki oleh masing-masing algoritma.

Analisis perbandingan antara SHA-1 dan SHA-256 menjadi relevan untuk dilakukan guna memahami kelebihan dan kekurangan dari kedua algoritma tersebut. Perbandingan ini dapat dilihat dari beberapa aspek, seperti kecepatan proses hashing, panjang output, tingkat keamanan terhadap serangan, serta efisiensi penggunaan dalam aplikasi berbasis PHP. Dengan adanya analisis ini, diharapkan dapat memberikan gambaran yang jelas bagi pengembang dalam menentukan algoritma hash yang наиболее sesuai dengan kebutuhan sistem yang dikembangkan. Selain itu, penelitian ini juga menjadi penting karena masih terdapat sistem lama (legacy systems) yang menggunakan SHA-1 sebagai mekanisme keamanan. Padahal, berbagai lembaga standar keamanan telah merekomendasikan untuk tidak lagi menggunakan SHA-1 dalam aplikasi baru karena kerentanannya. Oleh karena itu, diperlukan pemahaman yang mendalam mengenai perbedaan kedua algoritma ini agar proses migrasi ke algoritma yang lebih aman, seperti SHA-256, dapat dilakukan dengan baik tanpa mengganggu kinerja sistem secara signifikan.

Dalam penelitian ini, bahasa pemrograman PHP dipilih sebagai alat implementasi karena merupakan salah satu bahasa yang secara luas digunakan dalam pengembangan aplikasi web. PHP memiliki kemudahan dalam integrasi dengan berbagai sistem serta menyediakan fungsi-fungsi kriptografi yang cukup lengkap. Dengan menggunakan PHP, analisis terhadap performa dan hasil hashing dari SHA-1 dan SHA-256 dapat dilakukan secara langsung melalui pengujian terhadap berbagai jenis data input. Tujuan dari penelitian ini adalah untuk melakukan analisis

DOI: XXXXXXXXXX

perbandingan antara algoritma hash SHA-1 dan SHA-256 dalam implementasi menggunakan bahasa pemrograman PHP. Analisis ini meliputi aspek keamanan, performa, serta efisiensi dalam penggunaannya. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam bidang keamanan informasi, khususnya dalam pemilihan algoritma hash yang tepat untuk melindungi data dalam aplikasi berbasis web.

Dengan demikian, penelitian ini diharapkan tidak hanya memberikan pemahaman teoritis mengenai algoritma hash, tetapi juga memberikan gambaran praktis mengenai implementasinya dalam dunia nyata. Hal ini menjadi penting mengingat semakin meningkatnya ancaman keamanan di dunia digital, sehingga diperlukan solusi yang efektif dan efisien dalam menjaga integritas serta kerahasiaan data.

2 METODE PENELITIAN

Penelitian ini menggunakan pendekatan studi literatur (literature review) yang diperkuat dengan pengujian langsung (eksperimen) menggunakan bahasa pemrograman PHP. Kombinasi kedua pendekatan ini digunakan untuk memberikan hasil analisis yang lebih komprehensif, tidak hanya berdasarkan teori dari berbagai sumber ilmiah, tetapi juga berdasarkan implementasi nyata dalam lingkungan pemrograman.

Metode studi literatur digunakan sebagai pendekatan utama untuk mengkaji perbandingan algoritma hash SHA-1 dan SHA-256. Melalui metode ini, peneliti melakukan penelusuran, pengumpulan, serta analisis berbagai sumber ilmiah yang relevan seperti jurnal nasional dan internasional, buku referensi kriptografi, artikel ilmiah, prosiding, serta dokumentasi resmi dari lembaga standar seperti NIST dan IETF. Studi literatur ini bertujuan untuk memperoleh pemahaman mendalam mengenai konsep dasar fungsi hash, karakteristik SHA-1 dan SHA-256, serta perkembangan keamanan kriptografi modern.

Tahapan studi literatur dimulai dari identifikasi masalah, yaitu perbedaan tingkat keamanan dan performa antara SHA-1 dan SHA-256. Selanjutnya dilakukan pengumpulan referensi dari berbagai database ilmiah seperti Google Scholar, IEEE Xplore, dan ScienceDirect dengan mempertimbangkan relevansi topik, kredibilitas sumber, serta tahun publikasi yang relatif terbaru. Literatur yang telah dikumpulkan kemudian dianalisis secara deskriptif untuk membandingkan aspek-aspek penting seperti panjang output hash, tingkat keamanan terhadap collision attack, serta efisiensi algoritma.

Selain studi literatur, penelitian ini juga menggunakan metode pengujian langsung (eksperimen) menggunakan PHP untuk memperkuat hasil analisis teoritis. Pengujian dilakukan dengan mengimplementasikan algoritma SHA-1 dan SHA-256 menggunakan fungsi bawaan PHP, yaitu sha1() dan hash('sha256', ...). Pengujian ini bertujuan untuk melihat secara langsung perbedaan hasil output, panjang hash, serta waktu eksekusi kedua algoritma dalam lingkungan pemrograman nyata.

Data pengujian diperoleh dengan memberikan beberapa input berupa string dengan panjang yang berbeda, kemudian mencatat hasil hash yang dihasilkan serta waktu proses menggunakan fungsi microtime(true) pada PHP. Hasil pengujian ini tidak dimaksudkan sebagai eksperimen keamanan tingkat lanjut, tetapi sebagai validasi empiris sederhana untuk mendukung temuan dari studi literatur.

Setelah data dari studi literatur dan hasil pengujian terkumpul, dilakukan tahap analisis secara deskriptif-komparatif. Analisis ini digunakan untuk membandingkan kedua algoritma dari sisi teori dan implementasi. Aspek yang dianalisis meliputi tingkat keamanan, kompleksitas algoritma, panjang output hash, serta performa komputasi berdasarkan hasil pengujian PHP.

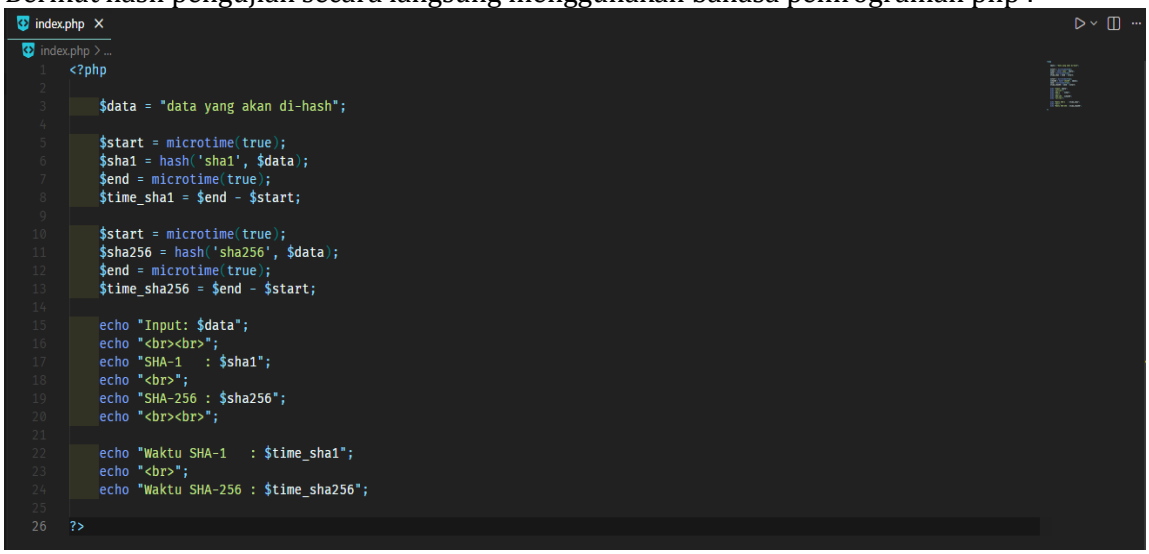
Dengan pendekatan gabungan ini, penelitian tidak hanya bersifat teoritis, tetapi juga memiliki dasar empiris yang diperoleh dari implementasi langsung. Hal ini diharapkan dapat memberikan gambaran yang lebih akurat mengenai perbandingan SHA-1 dan SHA-256, khususnya dalam konteks penggunaan pada bahasa pemrograman PHP serta relevansinya dalam pengembangan sistem keamanan modern.

DOI: XXXXXXXXXX

3 HASIL DAN PEMBAHASAN

A. Hasil

Berikut hasil pengujian secara langsung menggunakan bahasa pemrograman php :



Gambar 1 Source Code Hash SHA-1 dan SHA-256

Source code pada penelitian ini ditunjukkan pada Gambar 1 yang menggunakan bahasa pemrograman PHP sebagai alat untuk mengimplementasikan dan membandingkan algoritma hash SHA-1 dan SHA-256. Program ini dirancang untuk melakukan proses hashing terhadap sebuah data input yang telah ditentukan, yaitu berupa string “data yang akan di-hash”. Tujuan utama dari implementasi ini adalah untuk mengamati perbedaan hasil output serta waktu eksekusi dari kedua algoritma dalam kondisi yang sama.

Pada bagian awal program, ditentukan variabel $data yang berfungsi sebagai input utama yang akan diproses oleh kedua algoritma hash. Selanjutnya, proses pengujian dilakukan dengan mengukur waktu eksekusi menggunakan fungsi microtime(true) pada PHP. Fungsi ini digunakan untuk mendapatkan waktu dalam satuan mikrodetik sehingga perbedaan waktu pemrosesan dapat diamati dengan lebih akurat, meskipun dalam skala kecil.

Pengujian pertama dilakukan pada algoritma SHA-1 dengan memanggil fungsi hash('sha1', $data). Sebelum fungsi tersebut dieksekusi, waktu awal dicatat ke dalam variabel $start, kemudian setelah proses hashing selesai, waktu akhir dicatat ke variabel $end. Selisih antara waktu akhir dan waktu awal kemudian dihitung untuk memperoleh durasi eksekusi SHA-1, yang disimpan dalam variabel $time_sha1. Proses yang sama kemudian diterapkan pada algoritma SHA-256 menggunakan fungsi hash('sha256', $data). Hasil pengukuran waktu eksekusi disimpan dalam variabel $time_sha256. Dengan pendekatan ini, penelitian dapat membandingkan secara langsung efisiensi waktu pemrosesan antara kedua algoritma hash tersebut. Setelah proses hashing selesai, program menampilkan hasil output berupa nilai hash dari SHA-1 dan SHA-256 beserta waktu eksekusinya masing-masing. Output ini ditampilkan menggunakan fungsi echo dalam format HTML sederhana agar hasil dapat dilihat secara langsung melalui browser.

Secara keseluruhan, source code ini tidak hanya berfungsi sebagai implementasi teknis algoritma hash, tetapi juga sebagai alat eksperimen untuk membandingkan performa SHA-1 dan SHA-256 dalam lingkungan pemrograman PHP. Dari hasil implementasi ini, peneliti dapat mengamati perbedaan karakteristik kedua algoritma, baik dari sisi panjang output maupun waktu pemrosesan, yang kemudian digunakan sebagai dasar analisis dalam penelitian.

Berikut adalah Tabel 1 merupakan hasil pengujian berulang sebanyak 10 kali percobaan terhadap algoritma hash SHA-1 dengan data uji “data yang akan di-hash”.

Tabel 1 Hasil Pengujian Hash SHA-1

Hasil Enkripsi	Waktu
87192190d6fee991859052767e520368415d1d30	5.00

DOI: XXXXXXXXXX

87192190d6fee991859052767e520368415d1d30	5.00
87192190d6fee991859052767e520368415d1d30	9.05
87192190d6fee991859052767e520368415d1d30	5.00
87192190d6fee991859052767e520368415d1d30	5.00
87192190d6fee991859052767e520368415d1d30	6.19
87192190d6fee991859052767e520368415d1d30	5.96
87192190d6fee991859052767e520368415d1d30	5.00
87192190d6fee991859052767e520368415d1d30	6.19
87192190d6fee991859052767e520368415d1d30	5.0

Hasil pengujian menunjukkan bahwa algoritma SHA-1 menghasilkan nilai hash yang konsisten pada setiap percobaan, yaitu:

87192190d6fee991859052767e520368415d1d30

Hal ini menunjukkan bahwa SHA-1 bersifat deterministik, di mana input yang sama akan selalu menghasilkan output hash yang sama tanpa adanya perbedaan. Dari sisi waktu eksekusi, SHA-1 menunjukkan hasil yang relatif stabil namun memiliki variasi kecil pada setiap percobaan. Waktu tercepat yang tercatat berada pada kisaran 5.00 mikrodetik, sedangkan waktu terlama mencapai 9.05 mikrodetik. Rata-rata waktu eksekusi SHA-1 berada pada kisaran sekitar 5–6 mikrodetik, meskipun terdapat beberapa lonjakan kecil pada percobaan tertentu. Variasi ini dapat dipengaruhi oleh kondisi sistem saat pengujian berlangsung, seperti beban CPU, proses latar belakang, serta mekanisme caching pada server PHP.

Berikut adalah Tabel 2 merupakan hasil pengujian berulang sebanyak 10 kali percobaan terhadap algoritma hash SHA-256 dengan data uji “data yang akan di-hash”.

Tabel 2 Hasil Pengujian Hash SHA-1

Hasil	**Waktu**
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	2.86
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	2.14
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.99
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90
dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf	1.90

Pada algoritma SHA-256, hasil pengujian juga menunjukkan konsistensi output hash pada seluruh percobaan, yaitu:

dfcee2a725624203823b9d8fb84ac9f99c5c891506f1ecbd597c9fcacd65bebf

Sama seperti SHA-1, SHA-256 juga bersifat deterministik dan selalu menghasilkan output yang sama untuk input yang identik. Dari sisi waktu eksekusi, SHA-256 menunjukkan hasil yang relatif stabil dengan kisaran waktu yang lebih kecil dibandingkan SHA-1. Waktu tercepat berada pada 1.90 mikrodetik, sedangkan waktu terlama mencapai 2.86 mikrodetik. Secara umum, rata-rata waktu eksekusi SHA-256 berada di sekitar 1.9–2.1 mikrodetik.

Berdasarkan hasil kajian dari berbagai literatur kriptografi, algoritma SHA-1 dan SHA-256 merupakan bagian dari keluarga Secure Hash Algorithm yang dirancang untuk menghasilkan nilai hash sebagai representasi unik dari suatu data. Fungsi utama dari algoritma hash ini adalah memastikan integritas data, yaitu mendeteksi apakah suatu data telah mengalami perubahan atau tidak. Dari berbagai sumber yang dikaji, SHA-1 merupakan algoritma hash yang

menghasilkan nilai hash sepanjang 160-bit. Pada awalnya SHA-1 dianggap cukup kuat dan banyak digunakan dalam berbagai sistem keamanan seperti sertifikat digital, tanda tangan elektronik, serta validasi integritas file. Namun, perkembangan penelitian kriptografi menunjukkan bahwa SHA-1 mengalami penurunan tingkat keamanan yang signifikan. Hal ini disebabkan oleh ditemukannya metode collision attack yang memungkinkan dua input berbeda menghasilkan nilai hash yang sama. Temuan ini membuat SHA-1 tidak lagi direkomendasikan untuk penggunaan sistem keamanan modern.

Sementara itu, SHA-256 yang merupakan bagian dari keluarga SHA-2 menghasilkan nilai hash sepanjang 256-bit. Berdasarkan literatur dari NIST (National Institute of Standards and Technology), SHA-256 dirancang dengan struktur yang lebih kompleks dan ruang hash yang lebih besar dibandingkan SHA-1, sehingga memberikan tingkat keamanan yang lebih tinggi. Hingga saat ini SHA-256 masih digunakan secara luas dalam berbagai sistem modern seperti blockchain, SSL/TLS, digital signature, dan sistem autentikasi data. Dalam implementasinya pada bahasa pemrograman PHP, kedua algoritma ini telah didukung secara native melalui fungsi hash bawaan. Hal ini memungkinkan pengembang untuk mengimplementasikan SHA-1 maupun SHA-256 tanpa perlu membuat algoritma dari awal, cukup dengan menentukan parameter algoritma yang digunakan.

B. Pembahasan

Berdasarkan hasil sintesis literatur, perbedaan utama antara SHA-1 dan SHA-256 terletak pada tingkat keamanannya. SHA-1 telah terbukti memiliki kelemahan kriptografis yang serius, terutama terkait collision resistance. Collision attack yang berhasil ditemukan terhadap SHA-1 menunjukkan bahwa algoritma ini tidak lagi mampu memenuhi standar keamanan modern. Sebaliknya, SHA-256 memiliki ketahanan yang jauh lebih kuat terhadap serangan tersebut. Dengan panjang hash 256-bit, jumlah kemungkinan output yang dihasilkan jauh lebih besar dibandingkan SHA-1. Hal ini membuat proses pencarian collision secara brute force menjadi secara komputasi sangat tidak layak dilakukan dengan teknologi saat ini. Oleh karena itu, secara kriptografis SHA-256 dianggap masih aman dan direkomendasikan oleh standar keamanan internasional. Dari sisi performa, beberapa literatur menunjukkan bahwa SHA-1 memiliki kecepatan komputasi yang sedikit lebih tinggi dibandingkan SHA-256 karena proses komputasinya lebih sederhana dan panjang bit yang dihasilkan lebih kecil. Namun, perbedaan performa tersebut dalam praktiknya tidak terlalu signifikan, terutama pada perangkat komputasi modern yang telah memiliki kemampuan pemrosesan tinggi.

Dalam konteks penggunaan PHP pada aplikasi web, selisih waktu eksekusi antara SHA-1 dan SHA-256 tidak memberikan dampak besar terhadap kinerja sistem secara keseluruhan, kecuali pada sistem yang memproses data dalam jumlah sangat besar secara terus-menerus. Perbedaan panjang output menjadi faktor penting dalam membandingkan kedua algoritma ini. SHA-1 menghasilkan 160-bit hash, sedangkan SHA-256 menghasilkan 256-bit hash. Perbedaan ini bukan hanya bersifat kuantitatif, tetapi juga berdampak langsung pada tingkat keamanan. Semakin panjang nilai hash, maka semakin besar ruang kemungkinan kombinasi yang harus diuji oleh penyerang untuk menemukan collision. Dengan demikian, SHA-256 memiliki tingkat keamanan eksponensial yang lebih tinggi dibandingkan SHA-1.

Berdasarkan literatur terbaru, penggunaan SHA-1 saat ini telah ditinggalkan oleh banyak organisasi keamanan internasional. Bahkan beberapa browser modern dan sistem operasi sudah menolak sertifikat digital yang masih menggunakan SHA-1. Sebaliknya, SHA-256 telah menjadi standar de facto dalam berbagai sistem keamanan modern. Penggunaannya tidak hanya terbatas pada web security, tetapi juga pada teknologi blockchain seperti Bitcoin, yang mengandalkan SHA-256 sebagai dasar mekanisme hashing-nya. Dalam konteks pengembangan aplikasi berbasis web menggunakan PHP, penggunaan SHA-256 lebih direkomendasikan dibandingkan SHA-1. Hal ini disebabkan oleh faktor keamanan yang lebih kuat serta dukungan luas dalam standar industri. Walaupun PHP masih menyediakan dukungan untuk SHA-1, penggunaannya dalam sistem baru tidak lagi dianggap aman. Praktik yang lebih baik adalah menggunakan SHA-256 atau bahkan fungsi password_hash() untuk penyimpanan password karena telah dilengkapi dengan algoritma yang lebih aman seperti bcrypt atau argon2. Dengan

demikian, implementasi algoritma hash tidak hanya bergantung pada kemudahan teknis, tetapi juga pada pertimbangan keamanan jangka panjang.

4 KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan melalui kombinasi studi literatur dan pengujian langsung menggunakan bahasa pemrograman PHP, dapat disimpulkan bahwa terdapat perbedaan yang jelas antara algoritma hash SHA-1 dan SHA-256, baik dari aspek keamanan maupun performa. Dari hasil implementasi, kedua algoritma sama-sama menghasilkan output yang konsisten untuk setiap input yang sama, sehingga keduanya bersifat deterministik. Namun, SHA-256 memiliki panjang hash yang lebih besar dibandingkan SHA-1, yaitu 256-bit dibandingkan 160-bit, sehingga memberikan tingkat keamanan yang lebih tinggi dalam menjaga integritas data. Hasil pengujian juga menunjukkan bahwa terdapat perbedaan waktu eksekusi antara kedua algoritma. Pada pengujian yang dilakukan, SHA-256 menunjukkan waktu pemrosesan yang lebih kecil dibandingkan SHA-1.

Meskipun demikian, perbedaan tersebut tidak terlalu signifikan dan tidak memberikan dampak besar dalam penggunaan pada aplikasi web skala umum. Berdasarkan hasil kajian literatur, SHA-1 telah terbukti memiliki kelemahan dalam aspek keamanan, khususnya terhadap serangan collision, sehingga tidak lagi direkomendasikan untuk penggunaan pada sistem modern. Sebaliknya, SHA-256 masih dianggap aman dan telah banyak digunakan dalam berbagai sistem keamanan seperti SSL/TLS, tanda tangan digital, dan teknologi blockchain. Dengan demikian, dapat disimpulkan bahwa SHA-256 merupakan algoritma yang lebih unggul dibandingkan SHA-1, terutama dari sisi keamanan dan relevansi penggunaannya di era saat ini. Oleh karena itu, penggunaan SHA-256 lebih direkomendasikan dalam pengembangan sistem informasi berbasis PHP yang membutuhkan tingkat keamanan tinggi.

REFERENSI

[1] M. H. Qamaruzzaman *et al.*, “RANCANG BANGUN SISTEM PENGAMANAN DATA MEDIA FLASH DRIVE DENGAN ALGORITMA RIJNDAEL DAN HASH,” vol. 7, no. 2, pp. 29–35, 2025.

[2] A. H. Md, R. Dafi, A. Azhar, and I. S. Widiati, “Evaluasi Keamanan Penyimpanan Password Menggunakan,” pp. 1302–1305, 2025.

[3] N. H. Sari, P. Teknik, I. Universitas, and B. Darma, “Penerapan Teknik Digital Signature Dalam Pengamanan Piagam Penghargaan Menggunakan Algoritma SHA-1 dan RSA,” vol. 2, no. 3, pp. 55–67, 2024, doi: 10.47065/mis.v2i3.1465.

[4] D. N. Simatupang and E. Ardhianto, “PENAMBAHAN ASPEK INTEGRITAS INFORMASI PADA MODEL ENKRIPSI LIGHT WEIGHT PARALLEL ENCRYPTION WITH DIGIT ARITHMETIC OF COVERTEXT MENGGUNAKAN ALGORITMA SHA-1,” vol. 9, no. 3, pp. 4718–4726, 2025.

[5] A. Yahya, A. Kusyanti, P. H. Trisnawan, U. Brawijaya, and P. Korespondensi, “IMPLEMENTASI TIME-BASED ONE-TIME PASSWORD MENGGUNAKAN IMPLEMENTATION OF TIME-BASED ONE-TIME PASSWORD USING PHOTON ALGORITHM FOR TWO-FACTOR AUTHENTICATION,” vol. 12, no. 4, pp. 799–808, 2025.

[6] J. Teknologi, S. Informasi, K. Tgd, A. K. Hidayah, A. R. W. Mahfuzy, and M. Oki, “Implementasi Kombinasi Enkripsi Base64 Dengan Hashing Sha-1 Dan Md5 Pada Aplikasi Perpustakaan Universitas Muhammadiyah Bengkulu Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD,” vol. 6, pp. 694–703, 2023.

[7] A. Sha-, N. Sitorus, J. Sharon, G. Sinaga, and S. L. Samosir, “Analisis Kinerja Algoritma Hash pada Keamanan Data : Perbandingan,” vol. 2, no. 2, 2024.

[8] Y. A. Agustian *et al.*, “IMPLEMENTASI ALGORITMA SHA-256 PADA APLIKASI PENCATATAN,” vol. 10, no. 1, pp. 791–798, 2026.

[9] F. B. Pradana *et al.*, “IMPLEMENTASI BLOCKCHAIN MENGGUNAKAN ALGORITMA,” vol. 9, no. 8, pp. 1–7, 2025.

[10] V. Simangunsong, Y. R. Hutasoit, and D. Siallagan, “Analisis Terhadap Keamanan

Password Menggunakan Hash SHA-256," vol. 3, no. 1, 2025.
[11] A. Pengolahan and D. A. Untuk, "Journal of Computation Science And Artificial Intelligence," vol. 3, no. 1, pp. 25–37, 2026.
[12] J. Hutagalung, P. S. Ramadhan, S. J. Sihombing, and P. Korespondensi, "KEAMANAN DATA MENGGUNAKAN SECURE HASHING ALGORITHM (SHA) - 256 DAN RIVEST SHAMIR ADLEMAN (RSA) PADA DIGITAL SIGNATURE IMPLEMENTATION OF SECURE HASHING ALGORITHM (SHA) -256 AND RIVEST SHAMIR ADLEMAN (RSA) ON DIGITAL SIGNATURE," vol. 10, no. 6, 2023, doi: 10.25126/jtiik.2023107319.
[13] M. Winanda, S. Defrianti, W. Nabila, and H. Alfarizhi, "Implementasi Fungsi Hash dalam Kriptografi Modern untuk Enkripsi Data Satu Arah," vol. 1, no. 1, pp. 17–21, 2025.
[14] I. Artikel, "Implementasi QR Code dengan Algoritma Secure Hash Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) yang Ditingkatkan untuk Autentikasi Dokumen Digital," vol. 12, no. 1, pp. 11–22, 2024.
[15] M. Kombinasi and A. S.- Dan, "Implementasi Digital Signature Pada Faktur Dengan," vol. 5, pp. 112–124, 2022.