

# Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2

Nehemia Sitorus<sup>1</sup>, Joice Sharon Gabriella Sinaga<sup>2</sup>, Steven Lukas Samosir<sup>3</sup>

<sup>1,2,3</sup> Sarjana Terapan Teknologi Rekayasa Perangkat Lunak, Institut Teknologi Del  
strsnehemia88@gmail.com<sup>1</sup>

## Article Info

### Article history:

Received 4 Desember 2024

Revised 10 Desember 2024

Accepted 12 Desember 2024

### Keyword:

Data Security, Hash Algorithms  
Hash, SHA-256, SHA-3, Blake2

## ABSTRACT

Data security is an essential aspect in the dynamic digital age, including protection of data integrity, confidentiality and authentication. This research evaluates the performance and security of three popular hash algorithms SHA-256, SHA-3, and Blake2. Resistance to brute force attacks, avalanche effects, and processing time efficiency were analyzed. Results show that all algorithms have excellent resistance to brute force attacks, thanks to the 256-bit hash length. In terms of avalanche effect, SHA-3 shows higher consistency than the other algorithms, making it superior in maintaining data integrity. In terms of processing time efficiency, Blake2 performed best with the fastest hashing time, while SHA-256 offered balanced efficiency for general applications. SHA-3, although slower, has advantages in security and resistance to collision attacks. Overall, SHA-3 is recommended for applications with high security requirements, Blake2 is suitable for applications that prioritize efficiency, and SHA-256 remains relevant for applications that require a balance between speed and security.

This is an open access article under the CC Attribution 4.0 license.

## PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi di era digital saat ini telah menghasilkan volume data yang sangat besar dan kompleks. Dalam konteks ini, keamanan data menjadi aspek kritis yang tidak dapat diabaikan oleh organisasi, institusi, maupun individu. Seiring dengan meningkatnya ancaman siber yang semakin canggih, kebutuhan akan mekanisme perlindungan data yang handal dan efektif menjadi kebutuhan penting dalam sistem teknologi informasi modern [1].

Algoritma hash merupakan salah satu komponen fundamental dalam arsitektur keamanan siber. Fungsi utamanya adalah mengkonversi data input dengan panjang bervariasi menjadi output dengan panjang tetap, yang dikenal sebagai nilai hash. Karakteristik unik dari algoritma hash mencakup beberapa prinsip dasar: determinisme, kecepatan komputasi, resistensi terhadap kolisi, dan sifat satu arah yang tidak memungkinkan rekonstruksi data asli dari nilai hash [2].

Tiga algoritma hash yang menjadi fokus penelitian ini - SHA-256, SHA-3, dan Blake2 - mewakili generasi berbeda

dalam perkembangan teknologi kriptografis. SHA-256, bagian dari Secure Hash Algorithm (SHA) yang dikembangkan oleh National Security Agency (NSA), telah menjadi standar de facto dalam berbagai protokol keamanan [3]. SHA-3, yang dirancang melalui kompetisi NIST pada tahun 2012, menghadirkan arsitektur komputasi yang berbeda dengan mengadopsi konstruksi sponge. Sementara itu, Blake2 muncul sebagai solusi alternatif yang menawarkan performa tinggi dan keamanan yang kompetitif [4].

Penelitian ini telah berhasil mengidentifikasi dan mengevaluasi algoritma hash SHA-256, SHA-3, dan Blake2 berdasarkan parameter ketahanan terhadap serangan brute force, efek avalanche, dan waktu pemrosesan. Dengan pendekatan yang sistematis, penelitian ini memberikan wawasan strategis mengenai pemanfaatan algoritma hash dalam konteks keamanan data modern.

Melalui analisis mendalam, SHA-256 menunjukkan keandalannya sebagai standar yang mapan, terutama dalam aplikasi yang membutuhkan kompatibilitas tinggi dengan protokol keamanan konvensional. SHA-3, dengan arsitektur sponge yang inovatif, menawarkan perlindungan yang lebih

---

kuat terhadap serangan canggih, menjadikannya relevan bagi kebutuhan keamanan data masa depan. Di sisi lain, Blake2 membuktikan dirinya sebagai solusi optimal yang menggabungkan kecepatan tinggi dan keamanan kompetitif, ideal untuk sistem yang memprioritaskan efisiensi.

Penelitian ini tidak hanya memberikan rekomendasi praktis mengenai pemilihan algoritma hash berdasarkan skenario spesifik, tetapi juga menyoroti pentingnya mempertimbangkan keseimbangan antara keamanan, performa, dan kebutuhan aplikasi. Lebih jauh, hasil ini diharapkan dapat menjadi referensi bagi pengembangan lebih lanjut dalam bidang keamanan data serta berkontribusi pada wacana yang lebih luas tentang evolusi algoritma kriptografi dalam era digital.

Dalam pengujian terhadap serangan brute-force, SHA-256 menunjukkan ketahanan yang cukup baik, dengan waktu pemrosesan yang relatif cepat dibandingkan algoritma lainnya. Pengujian juga menunjukkan karakteristik avalanche effect, di mana perubahan kecil pada input menghasilkan perubahan signifikan pada output, yang merupakan atribut penting dalam algoritma hash [5].

SHA-256 pernah di uji untuk meningkatkan keamanan password pengguna, hasil dari pengujian menunjukkan nilai Avalanche effect sebesar 71% yang berarti hasil penyandian cukup baik dan aman dari serangan brute force. Selain itu, chipertext yang dihasilkan tidak dapat dipecahkan oleh serangan Rainbow Table.

SHA-3 pernah diteliti oleh peneliti yang membahas penerapan algoritma SHA-3 dalam proses autentikasi digital signature untuk piagam penghargaan. Proses ini menunjukkan bagaimana SHA-3 dapat digunakan untuk memastikan dan keaslian dokumen digital [6].

Blake2 pernah diuji pada penelitian yang dilakukan oleh peneliti yaitu menguji efisiensi dan keamanan dari Blake2 dalam konteks enkripsi data HTTP. Penelitian ini menunjukkan bahwa Blake2 sangat cepat, aman, serta efisiensi dibandingkan dengan beberapa algoritma lain dalam pengujian kecepatan hashing dan enkripsinya [7].

Tinjauan literatur menunjukkan bahwa meskipun SHA-256, SHA-3, dan Blake2 telah banyak diteliti, terdapat celah dalam analisis yang komprehensif. Penelitian sebelumnya cenderung berfokus pada satu algoritma dalam konteks aplikasi spesifik, seperti pengamanan password, autentikasi digital signature, atau enkripsi HTTP. Namun, belum ada studi yang secara sistematis membandingkan ketiga algoritma ini berdasarkan parameter yang seragam seperti efek avalanche, ketahanan terhadap brute force, dan efisiensi waktu pemrosesan.

Penelitian ini bertujuan untuk menganalisis kinerja algoritma hash dalam konteks keamanan data, dengan membandingkan SHA-256, SHA-3, Blake2, berdasarkan tiga parameter yaitu ketahanan terhadap serangan menggunakan brute force, efek avalanche, serta waktu pemrosesan. Dengan penelitian ini diharapkan dapat memberikan rekomendasi

tentang penggunaan algoritma mana yang paling sesuai berdasarkan analisis kinerja dan keamanan yang dilakukan.

## METODE

Tahap pengumpulan data ini dilakukan sesuai dengan kebutuhan yang terkait dengan tujuan penelitian, informasi yang dibutuhkan tergantung dari masalah yang dihadapi. Metode pengumpulan data menjadi penting karena bagaimana informasi yang terkumpul digunakan dan penjelasan apa yang dapat dihasilkan ditentukan oleh metodologi dan pendekatan analisis yang digunakan oleh peneliti. Semua Langkah dari proses pengumpulan data diperoleh melalui Studi Literatur

Studi literatur merupakan serangkaian kegiatan untuk mengumpulkan data dari berbagai sumber, termasuk bacaan dan Pustaka, yang berfungsi untuk mengelola bahan penelitian [8].

### A. Pengumpulan Data

Algoritma Hash merupakan algoritma yang berfungsi menghasilkan nilai hash yang bersifat deterministic, artinya input yang sama akan menghasilkan output yang sama, Jika terdapat perubahan kecil pada input, maka hash yang dihasilkan juga akan berubah secara signifikan[9] Di antara berbagai algoritma hash yang ada, SHA-256, SHA-3, Blake2 telah menjadi pilihan yang banyak digunakan untuk aplikasi keamanan.

SHA-256, yang dirancang oleh National Institute of Standards and Technology (NIST), merupakan bagian dari Secure Hash Algorithm (SHA) dalam keluarga SHA-2. Algoritma ini mengambil pesan dengan panjang lebih kecil dari  $2^{64}$  bit sebagai input dan menghasilkan output sepanjang 256 bit. SHA-256 menjadi salah satu fungsi hash yang paling aman dan paling banyak digunakan. Fungsi ini secara luas diaplikasikan untuk memastikan integritas data, penyimpanan kata sandi dalam bentuk terenkripsi, dan verifikasi transaksi [10], [11]. Penelitian menunjukkan bahwa SHA-256 memiliki ketahanan yang baik terhadap serangan brute-force, meskipun algoritma ini masih rentan terhadap serangan tertentu jika tidak diimplementasikan dengan benar. Dalam pengujian terhadap serangan brute-force, SHA-256 menunjukkan ketahanan yang cukup baik, dengan waktu pemrosesan yang relatif cepat dibandingkan algoritma lainnya. Pengujian juga menunjukkan karakteristik avalanche effect, di mana perubahan kecil pada input menghasilkan perubahan signifikan pada output, yang merupakan atribut penting dalam algoritma hash.

Algoritma SHA-3 adalah algoritma hash terbaru yang diangkat oleh NIST pada tahun 2015. Berbeda dengan SHA-256, SHA-3 menggunakan struktur sponge yang memberikan fleksibilitas lebih dalam ukuran output. Penelitian menunjukkan bahwa SHA-3 memiliki kinerja yang lebih baik dalam hal keamanan dibandingkan dengan SHA-1 dan

menunjukkan keunggulan dalam pengujian avalanche effect [12].

Sebuah penelitian menemukan bahwa SHA-3 menunjukkan ketahanan lebih tinggi terhadap serangan kolisi dibandingkan SHA-1, meskipun dalam beberapa pengujian waktu pemrosesan, SHA-3 lebih lambat akibat kompleksitas algoritmanya. Namun, hasil pengujian menunjukkan bahwa meskipun waktu pemrosesan lebih lama, efek keamanan SHA-3 jauh lebih signifikan. SHA-3 memiliki struktur yang lebih kompleks dibandingkan dengan SHA-256 yang dapat mengakibatkan waktu pemrosesan lebih lama. Algoritma Blake2 merupakan algoritma hash yang dirancang lebih cepat daripada SHA-256 dengan keamanan tinggi. Blake2 memiliki dua jenis yaitu :Blake2b untuk aplikasi 64-bit dan Blake2s untuk aplikasi 32-bit. Penelitian menunjukkan bahwa Blake2 tidak hanya cepat tetapi juga memiliki tingkat keamanan yang sebanding dengan SHA-3 dan SHA-256 [13].

Dalam pengujian, Blake2 menunjukkan waktu pemrosesan yang lebih cepat dibandingkan dengan SHA-256 dan SHA-3, menjadikan Blake2 lebih menarik digunakan untuk aplikasi yang membutuhkan kecepatan.

Brute Force adalah sebuah metode pencarian atau pemecahan masalah yang bekerja dengan mencoba setiap kemungkinan solusi secara sistematis hingga ditemukan solusi yang benar atau optimal. Dalam konteks keamanan komputer, metode ini sering digunakan untuk mencoba setiap kombinasi kata sandi atau kunci enkripsi hingga mendapatkan yang sesuai [14].

Avalanche Effect adalah sebuah karakteristik penting dalam fungsi hash kriptografis, di mana perubahan kecil pada input, seperti mengubah satu bit, menghasilkan perubahan signifikan pada output hash. Dalam istilah praktis, efek ini berarti bahwa lebih dari 50% bit dalam nilai hash akan berubah akibat modifikasi kecil pada input. Properti ini dianggap esensial untuk memastikan keamanan, karena mencegah pola-pola tertentu yang dapat dimanfaatkan dalam serangan seperti collision attack, length extension attack, atau preimage attack [15].

Time processing adalah waktu yang diperlukan oleh suatu sistem atau proses untuk menyelesaikan tugas tertentu, mulai dari tahap awal hingga tahap akhir. Waktu pemrosesan ini mencerminkan efisiensi dan kecepatan dalam pengolahan data, algoritma, atau sistem yang digunakan. Dalam konteks penelitian dan pengembangan teknologi, time processing sering digunakan sebagai indikator keberhasilan untuk mengukur performa, di mana pengurangan waktu pemrosesan menunjukkan peningkatan efisiensi dan produktivitas [16].

### *B. Pengaturan Pengujian*

Proses penelitian diawali dengan pengumpulan data yang akan digunakan sebagai input bagi ketiga algoritma hash. Data ini kemudian diproses menggunakan implementasi algoritma hash yang dijalankan dalam lingkungan sistem yang terstandarisasi. Lingkungan tersebut terdiri dari perangkat keras berprosesor AMD Ryzen 7, memori RAM sebesar 16 GB, dan sistem operasi Windows 10. Untuk

mendukung implementasi algoritma, digunakan pustaka Python hashlib untuk SHA-256 dan SHA-3, serta pustaka pycryptodome untuk Blake2 [17].

### *C. Implementasi Algoritma Hash*

SHA-256 diimplementasikan menggunakan pustaka hashlib. SHA-3 Menggunakan pustaka hashlib untuk fungsi SHA-3 yang tersedia sejak Python 3.6. Blake2 Diimplementasikan menggunakan pustaka pycryptodome, yang mendukung fungsi Blake2 dengan optimalisasi performa.

Pengujian algoritma dilakukan dengan beberapa pendekatan. Pertama, untuk mengukur kinerja algoritma, dilakukan pengujian waktu pemrosesan. Dalam pengujian ini, setiap algoritma dijalankan untuk memproses dataset dengan berbagai ukuran, dan waktu hashing diukur secara presisi menggunakan pustaka Python seperti timeit. Untuk menjaga konsistensi, setiap pengujian diulang sebanyak 10 kali, dan hasilnya dianalisis secara statistik untuk mendapatkan rata-rata, deviasi standar, dan rentang waktu pemrosesan.

### *D. Pengujian Kinerja*

Pengujian kinerja dilakukan dengan mengukur waktu hashing untuk setiap dataset dengan ukuran yang telah ditentukan. Waktu pemrosesan diukur menggunakan pustaka timeit dengan tingkat presisi dalam milidetik. Setiap pengujian diulang sebanyak 10 kali untuk setiap ukuran dataset, dan hasil pengujian dianalisis secara statistik. Analisis ini meliputi rata-rata waktu pemrosesan, deviasi standar, serta rentang waktu hashing untuk setiap algoritma.

### *E. Analisis Keamanan*

Keamanan algoritma dievaluasi berdasarkan dua parameter utama: efek avalanche dan ketahanan terhadap serangan brute force. Untuk mengukur efek avalanche, sebuah input awal dipilih secara acak dan dimodifikasi dengan perubahan kecil, seperti mengganti satu karakter atau membalikkan satu bit. Output hash dari input awal dan yang telah dimodifikasi dibandingkan untuk menghitung persentase bit yang berubah. Sementara itu, ketahanan terhadap brute force dianalisis secara teoritis dengan memperhitungkan jumlah kemungkinan kombinasi hash pada panjang output 256 bit, serta melalui simulasi dengan input acak.

## **HASIL DAN PEMBAHASAN**

Penelitian ini bertujuan untuk mengevaluasi performa algoritma hash SHA-256, SHA-3, dan Blake2 menggunakan pendekatan kuantitatif. Tiga parameter utama yang diukur dalam penelitian ini adalah ketahanan terhadap serangan brute force, efek avalanche, dan efisiensi waktu pemrosesan. Penelitian dilakukan dengan pengujian sistematis menggunakan input yang bervariasi untuk mengukur performa masing-masing algoritma dalam kondisi yang terkontrol.

### *A. Analisis Keamanan*



Iterasi ke 7	test12 34	testq234	937e8d5fb4 8bd494953 6cd65b8d3 5c426b80d 2f830c5c30 8e2cdec422 ae2244	9e7ae7581f fcf26c04fb 2a88e1121 1b6b65a2f d633c3bde 13247b74b a9b66bf0	<b>120</b>	Iterasi ke 4	test12 34	test123n	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	4f144915fc 7b819fd40 d8d22469b fad44880d 1b96a66b8 046d2127e 6d7b9cfb7	<b>108</b>	
Iterasi ke 8	test12 34	tGst1234	937e8d5fb4 8bd494953 6cd65b8d3 5c426b80d 2f830c5c30 8e2cdec422 ae2244	4fc4ee4763 c6317ecda 36852c216 f3b685c8e5 ef75e5d2ef 11be87736 8af6ecc	<b>124</b>	Iterasi ke 5	test12 34	test1V34	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	8dc6005d5 7c4756309 e5f50f4b39 4b3c9930b 0156be181 35ffff55d7 4176ea270f 2	<b>127</b>	
Iterasi ke 9	test12 34	test123D	937e8d5fb4 8bd494953 6cd65b8d3 5c426b80d 2f830c5c30 8e2cdec422 ae2244	e3c4b93f6c c3a752ca7 1b41f5f812 86723b526 3e1aff413e 51f1bdd3d 75d80a	<b>115</b>	Iterasi ke 6	test12 34	testG234	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	4e646c569 39a622c78 81444c4cd 75bd655f1 4f55b306d 947b9bc68 b6cf57478 1	<b>134</b>	
Iterasi ke 10	test12 34	test1F34	937e8d5fb4 8bd494953 6cd65b8d3 5c426b80d 2f830c5c30 8e2cdec422 ae2244	f1422d7ef8 24f99d607 e649a406a 70a63a0d2 573eaac8c5 bb541cb9b 4d82998	<b>122</b>	Iterasi ke 7	test12 34	ttst1234	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	e6531be27 bc63349ffc 145993c22 c09364357 074865f4f9 0a71c2e77 bdc56bae	<b>139</b>	
SHA-3	Iterasi ke 1	test12 34	test1934	0a8c939670 35827a6bce 55af0f4404 1dea04a3fff ff2f48d0e1 65a969b8b 8e97	45d1fcd36 67aea47c6 31a418679 c47b71abd 674c671a4 026c24f14 6542e298d	<b>132</b>	Iterasi ke 8	test12 34	teIt1234	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	ff9e82f4d0 287376a47 308a00587 656a3e92d 4d4805d61 9383733eb c4436407c b	<b>141</b>
Iterasi ke 2	test12 34	kest1234	0a8c939670 35827a6bce 55af0f4404 1dea04a3fff ff2f48d0e1 65a969b8b 8e97	05a45baf2f b28e60505 30d8bf5aad 13c7b72d7 d9c3c3712 2375a695b 2ae302	<b>129</b>	Iterasi ke 9	test12 34	testg234	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	10c2297c7 291286ce2 1346611b2 a35a41c2c 7f413f3e2f 97739d1c5 a9372384	<b>128</b>	
Iterasi ke 3	test12 34	Sest1234	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	036438e54 242a246d1 6ad6650e4 9cc295bef9 b756d2f66 2215955fcc f6d32115	<b>126</b>	Iterasi ke 10	test12 34	test123j	0a8c939670 3583270a6 bec55af0f4 4041dea04a 3fffffff2f4 80e0165a96 9b8b9e7	3f3640fd18 88c1ea278 2c98592c4 375379eac c88710f29 267a6978d 68723275	<b>121</b>	

---

Blake -2	Iterasi ke 1	test12 34	test1j34	833e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	9f92015ee8 d475382fb 232b09718 c79c15af39 dc062828c 4a1e47458 851166792 e2ff292001 15512d6e4 1f41612b0 9be0f71e00 150bebe67 4a9f52bb5 55a84	<b>237</b>
-------------	-----------------	--------------	----------	--	--	------------

Iterasi ke 2	test12 34	test123n	833e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	c40288801 82bdc740a 3c3a1181c a41b22bee af2bec99c7 a7f144485 08330d400 6c31715a0 6e692eef87 2a018a1ed 6e6b2b418 2488ec1c1 3109f8822 d278924	<b>267</b>
-----------------	--------------	----------	--	--	------------

Iterasi ke 3	test12 34	tesl1234	833e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	dd8cdf8d3 744c1b22a 9c4f16254 152188255 992e5a9ef8 8e4c46aae1 c5379096f 19943a3c4 c8bd7c749 5e67721e9 609c056c2 144c3d3fe4 fb3e12887 11dc	<b>255</b>
-----------------	--------------	----------	--	--	------------

Iterasi ke 4	test12 34	test1224	833e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	a35b27b6e ad292e5f51 193127f02 791c99443 662096e3e c47cb3602 482d0217b 21110da7b 2017726bf ad9448330 99c773922 74ea19549 07daeb078 86357	<b>251</b>
-----------------	--------------	----------	--	---	------------

---

Iterasi ke 5	test12 34	test1234	833e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	833e009ee 96041e1ec c8cdfedcab e2ac2ebe24 e326abb17 1a48017d0 9d32c7b23 43dfad71f0 8da699ed9 6e5570969 d053920f7 2218a2cc9 1f17b298b 10ef95	<b>0</b>
-----------------	--------------	----------	--	--	----------

Iterasi ke 6	test12 34	Oest123 4	a33e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	e2acc46f4a a29176adc 00c7384c3 b1841f8e8 079a4b084 354c59895 5a67812f2 744308e32 776130484 4f8992777 2647c5847 57c29b569 61544c2f9 6c1	<b>276</b>
-----------------	--------------	--------------	--	---	------------

Iterasi ke 7	test12 34	test123C	a33e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	c6e7181ef8 53f42d7c2 65cac4f41d 57cd971c0 203591435 a572b4be2 748864f31 1e1ce43b3 4fc801e8b1 74c46c816f 96bce3368 588bfc677a f82a4525c2 a1	<b>256</b>
-----------------	--------------	----------	--	---	------------

Iterasi ke 8	test12 34	testW23 4	a33e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	61198f8f79 59e5f7578 74a0d5112 0d8965e1a 9c82b4af61 3cde942d1 9037a80d8 ce083642e cf42488c0c e49690393 07895fcc6d 9987b4214 1ee39940b 0d	<b>262</b>
-----------------	--------------	--------------	--	---	------------

---

Iterasi ke 9	test1234	teCt1234	a33e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	ca36a2317 9dccc5839 5e2fbb8c20 6e50f973a8 741306915 384bfe659 5421fe1b5f c08848903 b6a92b4c5 1e47699a0 0fe26354e9 7016fc4fb6 743ba3a81 3	243
Iterasi ke 10	test1234	t9st1234	a33e009ee9 6041e1ecc8 cdfedcabe2 ac2ebe24e3 26abb171a4 8017d09d3 2c7b2343df ad71f08da6 99ed96e557 0969d0539 20f72218a2 cc91f17b29 8b10ef95	ec27db89b 46698f988 58425e18e 0d4d9b09a 65e455126 8121013d4 0463649d2 ce3e85a76 37f9967a6 c2f9ac1688 9d19ba6eb 69b40bbb2 c16622ee5 90ed75b	255

Hasil pengujian menunjukkan serangkaian pengujian menggunakan algoritma SHA-256 dan SHA-3. Input awal yang digunakan adalah string "test1234" dan "test123C/test123/test1230/test1234". Input diubah secara sistematis dengan mengganti satu karakter atau satu bit pada input. Misalnya, pada Gambar 1 input "test1234" diubah menjadi "testQXYZ", "testZXYQ", dan seterusnya.

Proses Hashing menunjukkan Input awal dan input yang telah dimodifikasi diproses menggunakan algoritma hash SHA-256. Hasil hash (nilai hash) dari input awal dan input yang dimodifikasi ditampilkan dalam bentuk heksadesimal.

Efek avalanche dihitung dengan membandingkan jumlah bit yang berubah antara output hash awal dan output hash setelah perubahan input. Pada Gambar 1, dapat dilihat bahwa ketika input diubah hanya satu karakter, hasil hash berubah secara signifikan. Misalnya, pada iterasi 1, hash awal adalah "937e8d5fbb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244", sedangkan hash setelah perubahan input menjadi "59f078e40cc8998c1107b9e9422d055f1a2ee99324e492c52b277cd84fb2b859". Selisih bit antara hash awal dan hash setelah perubahan input berkisar antara 48-55%, menunjukkan efek avalanche yang baik.

Analisis Hasilnya Perubahan kecil pada input, bahkan hanya satu karakter atau satu bit, menghasilkan perubahan yang signifikan pada output hash. Jumlah bit yang berubah antara output hash awal dan output hash setelah modifikasi input berada pada rentang 48-55%, mendekati karakteristik ideal efek avalanche (50%). Hal ini menunjukkan bahwa algoritma SHA-256 memiliki efek avalanche yang baik, di

mana perubahan kecil pada input akan menghasilkan perubahan yang signifikan pada output hash.

### C. Pengukuran waktu pemrosesan

Efisiensi waktu pemrosesan adalah parameter penting, khususnya untuk aplikasi yang membutuhkan hashing data secara real-time. Pengujian dilakukan untuk mengukur waktu hashing pada dataset dengan ukuran yang bervariasi.

TABEL II  
PENGUJIAN EFEK AVALANCHE

Ukuran Besar File	SHA-256	SHA-3	Blake-2
1 kb	0,00000...detik k	0,00000... detik	0,00000... detik
10 kb	0,00000... detik	0,9 detik	0,00000... detik
100 kb	0,9 detik	1 detik	1 detik
1 mb	0,00502 detik	0,01959 detik	0,01552 detik

Panjang hash ketiga algoritma sama-sama 256-bit, sehingga ketahanan brute force secara teoretis sama. Dengan asumsi kemampuan modern (1 triliun percobaan/detik), waktu brute force tetap tidak praktis (ribuan tahun untuk menemukan kecocokan).

Dataset input dibuat dengan ukuran berbeda, yaitu 1 KB, 10 KB, 100 KB, hingga 1 MB. Input data dihasilkan secara acak untuk memastikan hasil pengujian tidak bias terhadap pola tertentu. Setiap dataset diproses menggunakan algoritma SHA-256, SHA-3, dan Blake2 untuk menghasilkan nilai hash. Waktu hashing diukur dengan presisi milidetik menggunakan pustaka Python seperti time atau timeit. Setiap pengukuran diulang sebanyak 10 kali untuk memastikan hasil yang konsisten dan mengurangi pengaruh fluktuasi sistem. Hasil Pengujian: SHA-256: Efisiensi waktu hashing meningkat secara linier dengan ukuran input. Untuk input 1 MB, waktu pemrosesan rata-rata adalah 0,00502 detik. SHA-3: Waktu hashing lebih lama dibandingkan dengan SHA-256 dan Blake2 karena kompleksitas algoritmanya. Input 1 MB membutuhkan waktu rata-rata 0,01959 detik. Blake2: Memiliki waktu hashing tercepat, terutama pada input besar. Input 1 MB hanya membutuhkan waktu rata-rata 0,01552 detik. Blake2 unggul dalam efisiensi waktu pemrosesan, diikuti oleh SHA-256, sementara SHA-3 menunjukkan waktu pemrosesan yang lebih lama karena struktur algoritmanya yang lebih kompleks.

## SIMPULAN

Penelitian ini menganalisis kinerja dan keamanan dari tiga algoritma hash, yaitu SHA-256, SHA-3, dan Blake2, berdasarkan tiga parameter utama: ketahanan terhadap serangan brute force, efek avalanche, dan efisiensi waktu pemrosesan. Berdasarkan hasil pengujian, ketiga algoritma menunjukkan ketahanan yang sangat baik terhadap serangan brute force, berkat panjang hash yang mencapai 256 bit, yang membuat serangan brute force menjadi tidak praktis dalam skala waktu yang wajar.

Dari segi efek avalanche, ketiga algoritma mampu menghasilkan perubahan output yang signifikan ketika terjadi perubahan kecil pada input. Namun, SHA-3 menunjukkan konsistensi yang lebih tinggi dalam menghasilkan efek avalanche, yang menjadikannya lebih unggul dalam menjaga integritas data terhadap perubahan input yang kecil.

Dalam hal efisiensi waktu pemrosesan, Blake2 menunjukkan performa terbaik, dengan waktu hashing tercepat dibandingkan dengan SHA-256 dan SHA-3. SHA-256 memberikan efisiensi yang cukup baik untuk aplikasi umum, sementara SHA-3, meskipun lebih lambat, menunjukkan keunggulan dalam hal keamanan dan ketahanan terhadap serangan kolisi.

Secara keseluruhan, SHA-3 unggul dalam aspek keamanan dan efek avalanche, menjadikannya pilihan terbaik untuk aplikasi dengan kebutuhan keamanan tinggi. Blake2 menawarkan kombinasi terbaik antara kecepatan dan keamanan, sehingga lebih cocok untuk aplikasi yang memprioritaskan efisiensi waktu pemrosesan. SHA-256 tetap menjadi pilihan yang relevan karena kesei

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dalam penyelesaian penelitian ini. Terima kasih khusus disampaikan kepada rekan-rekan dan dosen yang memberikan masukan berharga, serta institusi yang menyediakan fasilitas dan sumber daya yang mendukung penelitian ini. Semoga hasil penelitian ini bermanfaat bagi pengembangan keamanan data di masa depan.

## DAFTAR PUSTAKA

- [1] F. Kurniawan, A. Kusyanti, and H. Nurwarsito, "Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 9, pp. 803–812, 2017, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/247>
- [2] I. Malviya and T. Chetty, "International Journal on Recent and

- Innovation Trends in Computing and Communication: Performance and Limitation Review of Secure Hash Function Algorithm," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 7, no. 6, pp. 48–51, 2019, [Online]. Available: <http://www.ijritcc.org>
- [3] R. K. Dahal, J. Bhatta, and T. N. Dhamala, "Performance Analysis of Sha-2 and Sha-3 Finalists," *Int. J. Cryptogr. Inf. Secur.*, vol. 3, no. 3, pp. 1–10, 2013, doi: 10.5121/ijcis.2013.3301.
- [4] P. K. Yien, K. Malik, and S. N. Ramli, "Comparative Study on Hash Function Algorithms for Blockchain Technology," vol. 5, no. 1, pp. 16–33, 2024.
- [5] H. R. Ngemba, - Ifandi, S. Hendra, and I. G. N. A. K. Dwi Arsana, "Implementasi Enkripsi Data MD5 dan SHA-256 pada Sistem Informasi Peminjaman Buku Tanah," *Techno.Com*, vol. 22, no. 3, pp. 654–664, 2023, doi: 10.33633/tc.v22i3.8299.
- [6] A. Apriliani, N. A. Hasibuan, and D. P. Utomo, "Implementasi Algoritma SHA-3 Dan ElGamal Untuk Otentikasi Piagam Penghargaan Berbasis Digital Signature," *SNASTIKOM Semin. Nas. Teknol. Inf. Komun.*, vol. 9, no. 1, pp. 157–164, 2022.
- [7] B. D. Kurniawan, M. A. Rosid, I. A. Kautsar, and N. E. Pratama, "Rancang Bangun Library Web Token untuk Enkripsi HTTP Data Menggunakan Eksklusif-OR (XOR)," *Phys. Sci. Life Sci. and Engineering*, vol. 1, no. 1, p. 14, 2024, doi: 10.47134/pslse.v1i1.164.
- [8] M. Pusparani, "Faktor Yang Mempengaruhi Kinerja Pegawai (Suatu Kajian Studi Literatur Manajemen Sumber Daya Manusia)," *J. Ilmu Manaj. Terap.*, vol. 2, no. 4, pp. 534–543, 2021, doi: 10.31933/jimt.v2i4.466.
- [9] Pinkan Indriani Daulay and Yahfizham Yahfizham, "Penerapan Algoritma Pemrograman dalam Pembelajaran Ilmu Komputer," *J. Arjuna Publ. Ilmu Pendidikan, Bhs. dan Mat.*, vol. 1, no. 6, pp. 91–103, 2023, doi: 10.61132/arjuna.v1i6.297.
- [10] N. Alamgir, S. Nejjati, and C. Bright, "SHA-256 Collision Attack with Programmatic SAT," *CEUR Workshop Proc.*, vol. 3717, pp. 91–110, 2024.
- [11] M. DiNardi and D. Radhakrishnan, "SHA-256 Hash Function on Intel DE10 Lite FPGA," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 12, pp. 1175–1184, 2023, doi: 10.22214/ijraset.2023.57521.
- [12] A. Ismail, V. A. H. F., and A. T. F., "Sistem Tanda Tangan Digital Menggunakan SHA-3 dan ECDSA," *Unistek*, vol. 10, no. 2, pp. 84–93, 2023.
- [13] M. A. Fadhillah, L. Mulyarahim, and K. Nadira, "Algoritme Hashing Sha-512 Pada Sistem Halaman Sign Up Java," *TRIPLE A J. Pendidik. Teknol. Inf.*, vol. 2, no. 1, pp. 27–34, 2023.
- [14] S. A. Rahmah, "Efektifitas Penerapan Algoritma Brute Force dan Penyalahgunaannya Dalam Sistem Berbasis Web," *J. Comput. Digit. Bus.*, vol. 2, no. 3, pp. 112–119, 2023.
- [15] D. Upadhyay, N. Gaikwad, M. Zaman, and S. Sampalli, "Investigating the Avalanche Effect of Various Cryptographically Secure Hash Functions and Hash-Based Applications," *IEEE Access*, vol. 10, no. October, pp. 112472–112486, 2022, doi: 10.1109/ACCESS.2022.3215778.
- [16] N. Varela, C. Ospino, and O. B. P. Lezama, "Methodology for processing time series using machine learning," *Procedia Comput. Sci.*, vol. 175, pp. 659–664, 2020, doi: 10.1016/j.procs.2020.07.096.
- [17] A. H. Montiel, A. F. Martínez, and G. E. Jacinto, "Implementation of Password Hashing on Embedded Systems with Cryptographic Acceleration Unit," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 2, pp. 171–175, 2022, doi: 10.14569/IJACSA.2022.0130221.