Jurnal Quancom
Vol.2, No.2, Desember 2024, pp. 9~16
e-ISSN: 2988-0998
P-ISSN: 2988-1005

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

Nehemia Sitorus[1], Joice Sharon Gabriella Sinaga[2], Steven Lukas Samosir[3]
[1,2,3] Sarjana Terapan Teknologi Rekayasa Perangkat Lunak, Institut Teknologi Del
strsnehemia88@gmail.com[1]

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.

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 yag dilakukan oleh peneliti yaitu menguji efiesiensi dan kemanan 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

Pengujian ini bertujuan untuk mengukur seberapa besar perubahan pada hash ketika satu karakter atau satu bit pada

input diubah. Input awal adalah string acak dengan panjang 32 karakter, dihasilkan dari kombinasi huruf dan angka.

```python
# Fungsi untuk menghitung hash menggunakan SHA-256
def sha256_hash(input_data):
    return hashlib.sha256(input_data.encode()).hexdigest()

# Fungsi untuk menghitung hash menggunakan SHA-3
def sha3_hash(input_data):
    sha3 = SHA3_256.new()
    sha3.update(input_data.encode())
    return sha3.hexdigest()

# Fungsi untuk menghitung hash menggunakan Blake2
def blake2_hash(input_data):
    blake2 = BLAKE2b.new(digest_bits=256)  # Output Blake2 diatur ke 256-bit
    blake2.update(input_data.encode())
    return blake2.hexdigest()

# Target hash untuk dicocokkan (diambil dari hash input tetap "password123")
target_sha256 = sha256_hash("password123")
target_sha3 = sha3_hash("password123")
target_blake2 = blake2_hash("password123")

print(f"Hash target (SHA-256): {target_sha256}")
print(f"Hash target (SHA-3): {target_sha3}")
print(f"Hash target (Blake2): {target_blake2}")

# Panjang input yang akan diuji
input_lengths = [8, 16, 32, 64]

# Simulasi brute force
for algo_name, hash_function, target_hash in [
    ("SHA-256", sha256_hash, target_sha256),
    ("SHA-3", sha3_hash, target_sha3),
    ("Blake2", blake2_hash, target_blake2),
]:
```

Gambar 1. Code pengujian

```
Hash target (SHA-256): ef92b778bafe771e89245b89ecbc08a44a4e166c06659911881f383d4473e94f
Waktu teoretis untuk brute force: 3671743063080802666178524274123826886830384916383622234112.00 tahun

Simulasi brute force untuk panjang input: 8 karakter
Masih mencoba... Iterasi ke-1000000
Masih mencoba... Iterasi ke-2000000
Masih mencoba... Iterasi ke-3000000
Masih mencoba... Iterasi ke-4000000
Masih mencoba... Iterasi ke-5000000
Masih mencoba... Iterasi ke-6000000
Masih mencoba... Iterasi ke-7000000
Masih mencoba... Iterasi ke-8000000
Masih mencoba... Iterasi ke-9000000
Masih mencoba... Iterasi ke-10000000
Masih mencoba... Iterasi ke-11000000
Masih mencoba... Iterasi ke-12000000
Masih mencoba... Iterasi ke-13000000
Masih mencoba... Iterasi ke-14000000
Masih mencoba... Iterasi ke-15000000
Masih mencoba... Iterasi ke-16000000
Masih mencoba... Iterasi ke-17000000
Masih mencoba... Iterasi ke-18000000
Masih mencoba... Iterasi ke-19000000
Masih mencoba... Iterasi ke-20000000
Masih mencoba... Iterasi ke-21000000
Masih mencoba... Iterasi ke-22000000
Masih mencoba... Iterasi ke-23000000
```

Gambar 2. Pengujian keamanan SHA-256 terhadap Brute Force

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).

B. Pengujian efek Avalanche

Efek avalanche merupakan salah satu karakteristik penting dari fungsi hash dimana perubahan kecil pada input fungsi hash akan menghasilkan output yang sangat berbeda. Efek ini mengacu pada kemampuan algoritma untuk menghasilkan perubahan besar pada output hash ketika terdapat perubahan kecil pada input. Avalanche effect dihitung dengan membandingkan jumlah bit yang berubah antara output hash awal dan output hash setelah perubahan input.

TABEL I
PENGUJIAN EFEK AVALANCHE

Algoritma	Iterasi Ke-	Input Asli	Input di ubah	Hash Asli	Hash diubah	Bit Berbeda
SHA-256	Iterasi ke 1	test1234	tesQ1234	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	2c4eb9522cc592eea805358dc1d79dd7e12318041fb87f6f94a8f3fc7421b	**133**
	Iterasi ke 2	test1234	testf234	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	59f078b40cc8998c1107b9e94220d55f1ae2eef9324e402c52b27cdb4fb2d859	**132**
	Iterasi ke 3	test1234	test1234	0a8c93967035832 7a6bec55af0f44041dea04a3ffffff24f8d0e165a969b8b9e7	03b643b5942b2a46d1ebda650e49cc205baf9b756d2f66221595fcfd39211d5	**0**
	Iterasi ke 4	test1234	test123G	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	917c166bd4d36f074c57c5e9a9a8ff4b4db92440f521834ff15c6360f08614	**126**
	Iterasi ke 5	test1234	test123C	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	76967f0fb60400cd524f742e1420b3679d7e32abc4a57f8b17cb39ae38b49735	**134**
	Iterasi ke 6	test1234	test1s34	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	9a91187447f400c4e77be98acc9e497032d65cb9ca4347374cf4cdd69b775a81	**128**

	Iterasi ke 7	test1234	testq234	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	9e7ae7581ffcf26c04fb2a88e11211b6b65a2fd633c3bde13247b74ba9b66bf0	**120**
	Iterasi ke 8	test1234	tGst1234	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	4fc4ee4763c6317ecda36852c216f3b685c8e5ef75e5d2ef11be877368af6ecc	**124**
	Iterasi ke 9	test1234	test123D	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	e3c4b93f6cc3a752ca71b41f5f81286723b5263e1aff413e51f1bdd3d75d80a	**115**
	Iterasi ke 10	test1234	test1F34	937e8d5fb48bd4949536cd65b8d35c426b80d2f830c5c308e2cdec422ae2244	f1422d7ef824f99d607e649a406a70a63a0d2573eaac8c5bb541cb9b4d82998	**122**
SHA-3	Iterasi ke 1	test1234	test1934	0a8c93967035827a6bce55af0f44041dea04a3ffff2f48d0e165a969b8b8e97	45d1fcd3667aea47c631a418679c47b71abd674c671a4026c24f146542e298d	**132**
	Iterasi ke 2	test1234	kest1234	0a8c93967035827a6bce55af0f44041dea04a3ffff2f48d0e165a969b8b8e97	05a45baf2fb28e6050530d8bf5aad13c7b72d7d9c3c37122375a695b2ae302	**129**
	Iterasi ke 3	test1234	Sest1234	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	036438e54242a246d16ad6650e49cc295bef9b756d2f6622215955fccf6d32115	**126**
	Iterasi ke 4	test1234	test123n	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	4f144915fc7b819fd40d8d22469bfad44880d1b96a66b8046d2127e6d7b9cfb7	**108**
	Iterasi ke 5	test1234	test1V34	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	8dc6005d57c4756309e5f50f4b394b3c9930b0156be18135fffff55d74176ea270f2	**127**
	Iterasi ke 6	test1234	testG234	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	4e646c56939a622c7881444c4cd75bd655f14f55b306d947b9bc68b6cf574781	**134**
	Iterasi ke 7	test1234	ttst1234	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	e6531be27bc63349ffc145993c22c09364357074865f4f90a71c2e77bdc56bae	**139**
	Iterasi ke 8	test1234	teIt1234	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	ff9e82f4d0287376a47308a00587656a3e92d4d4805d619383733ebc4436407cb	**141**
	Iterasi ke 9	test1234	testg234	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	10c2297c7291286ce21346611b2a35a41c2c7f413f3e2f97739d1c5a9372384	**128**
	Iterasi ke 10	test1234	test123j	0a8c9396703583270a6bec55af0f44041dea04a3ffffffff2f480e0165a969b8b9e7	3f3640fd1888c1ea2782c98592c4375379eacc88710f29267a6978d68723275	**121**

Blake-2	Iterasi ke 1	test1234	test1j34	833e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	9f92015ee8d475382fb232b09718c79c15af39dc062828c4a1e47458851166792e2ff29200115512d6e41f41612b09be0f71e00150bebe674a9f52bb555a84	**237**
	Iterasi ke 2	test1234	test123n	833e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	c4028880182bdc740a3c3a1181ca41b22beeaf2bec99c7a7f14448508330d4006c31715a06e692eef872a018a1ed6e6b2b4182488ec1c13109f8822d278924	**267**
	Iterasi ke 3	test1234	tesl1234	833e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	dd8cdf8d3744c1b22a9c4f16254152188255992e5a9ef88e4c46aae1c5379096f19943a3c4c8bd7c7495e67721e9609c056c2144c3d3fe4fb3e1288711dc	**255**
	Iterasi ke 4	test1234	test1224	833e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	a35b27b6ead292e5f51193127f02791c99443662096e3ec47cb3602482d0217b21110da7b2017726bfad9448330 99c77392274ea1954907daeb07886357	**251**
	Iterasi ke 5	test1234	test1234	833e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	833e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	**0**
	Iterasi ke 6	test1234	Oest1234	a33e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	e2acc46f4aa29176adc00c7384c3b1841fbe8079a4b084354c598955a67812f2744308e32776130484 4f89927772647c584757c29b56961544c2f96c1	**276**
	Iterasi ke 7	test1234	test123C	a33e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	c6e7181ef853f42d7c265cac4f41d57cd971c0203591435a572b4be2748864f311e1ce43b34fc801e8b174c46c816f96bce3368588bfc677af82a4525c2a1	**256**
	Iterasi ke 8	test1234	testW234	a33e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	61198f8f7959e5f757874a0d51120d8965e1a9c82b4af613cde942d19037a80d8ce083642ecf42488c0ce4969039307895fcc6d9987b42141ee39940b0d	**262**

Iterasi ke 9	test1234	teCt1234	a33e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	ca36a23179dccc58395e2fbb8c206e50f973a8741306915384bfe6595421fe1b5fc08848903b6a92b4c51e47699a00fe26354e97016fc4fb6743ba3a813	**243**
Iterasi ke 10	test1234	t9st1234	a33e009ee96041e1ecc8cdfedcabe2ac2ebe24e326abb171a48017d09d32c7b2343dfad71f08da699ed96e5570969d053920f72218a2cc91f17b298b10ef95	ec27db89b46698f98858425e18e0d4d9b09a65e4551268121013d40463649d2ce3e85a7637f996e7a6c2f9ac16889d19ba6eb69b40bbb2c16622ee590ed75b	**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	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. Nejati, 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.