Analisis Perbandingan Avalanche Effect pada Fungsi Hash SHA-256 dan SHA3-256

Raga Fatwa Ikhwani - 13222001
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13222001@std.stei.itb.ac.id, [2]ragafatwa@gmail.com

Abstract*—Dilakukan analisis perbandingan karakteristik Avalanche Effect pada dua fungsi hash, yaitu SHA-256 dan SHA3-256. Avalanche Effect merupakan karakteristik dalam kriptografi, yaitu perubahan minimal pada input akan menghasilkan perubahan besar pada output dengan nilai ideal 50% perubahan pada bit output. Metode percobaan melibatkan pengujian eksperimental menggunakan bahasa pemrograman Python dengan memodifikasi 1 bit pada input secara acak (1-bit flip). Perbedaan bit pada output diukur menggunakan metode Hamming Distance dan dikonversi menjadi persentasi Avalanche Effect. Pengujian dilakukan secara berulang sebanyak 100 dan 1000 kali untuk memastikan validitas statistik. Hasil percobaan menunjukkan bahwa kedua fungsi hash memenuhi kriteria Avalanche Effect yang baik dengan nilai rata-rata mendekati 50%. Pada pengujian 100 kali, SHA-256 memiliki nilai rata-rata 49.73% dan SHA3-256 memiliki nilai rata-rata 49.94%. Pada pengujian 1000 kali, SHA-256 memiliki nilai rata-rata yang sama yaitu 49.74% dan SHA3-256 memiliki nilai yang sangat baik yaitu 49.99%. Perbedaan ini memperlihatkan keunggulan struktur yang digunakan pada SHA3-256 dalam hal konsistensi difusi bit.

Keywords*—Avalanche Effect, Hamming Distance, Merkle-Damgård, SHA-256, SHA3-256, Sponge Construction

I. Pendahuluan

Perkembangan teknologi informasi telah mendorong kebutuhan dalam mekanisme keamanan data. Fungsi hash dalam kriptografi memiliki peran penting dalam berbagai aplikasi seperti penyimpanan data, komunikasi digital, dan teknologi blockchain. Fungsi hash mampu mengubah data input dengan panjang yang bervariasi menjadi output dengan panjang tetap dan sulit untuk diprediksi dan dimanipulasi.

Salah satu karakteristik utama yang dimiliki oleh fungsi hash dalam kriptografi adalah Avalanche Effect. Avalanche Effect menggambarkan perubahan sekecil apapun pada data input, seperti perubahan satu bit, akan menyebabkan perubahan besar pada hasil/outputnya. Idealnya, perubahan satu bit pada input diharapkan dapat mengubah 50% bit pada output hash. Karakteristik ini penting untuk mencegah analisis pola dan serangan kriptanalisis.

SHA-256 dan SHA3-256 merupakan salah dua algoritma hash kriptografi dalam sistem keamanan modern. SHA-256 telah lama menjadi standar dalam berbagai aplikasi keamanan. Sedangkan SHA3-256 merupakan standar alternatif yang dapat memberikan tingkat keamanan yang lebih baik.

Kedua algoritma tersebut memiliki perbedaan struktur dan mekanisme. Oleh karena itu, percobaan ini bertujuan untuk melakukan analisis perbedaan Avalanche Effect antara SHA-256 dan SHA3-256. Analisis dilakukan dengan memodifikasi satu bit pada input dan mengamati perubahan pada hasil/output hash.

II. Landasan Teori

A. Fungsi Hash

Fungsi hash dalam kriptografi merupakan suatu fungsi matematis yang memetakan data masukan (input) dengan panjang yang bervariasi menjadi keluaran (output) dengan panjang yang tetap [1]. Fungsi hash dirancang agar bersifat deterministik namun sangat sulit untuk memperoleh Kembali input asli dari nilai hash yang dihasilkan.

Dalam kriptografi, fungsi hash banyak digunakan pada berbagai aplikasi keamanan, contohnya seperti penyimpanan *password*/kata sandi [1]. Tujuan utama fungsi hash yaitu menjamin perubahan sekecil apapun pada input dapat terdeteksi melalui perubahan nilai hash. Oleh karena itu, fungsi hash banyak digunakan untuk memastikan integritas suatu data.

B. Avalanche Effect

Avalanche Effect merupakan salah satu karakteristik dalam kriptografi yang pertama kali dikenalkan oleh Horst Feistel [2] pada cipher modern. Avalanche Effect menyatakan bahwa setiap perubahan kecil pada input, seperti perubahan 1 bit, harus menghasilkan perubahan besar dan tidak terprediksi pada outputnya [2]. Karakteristik ini bertujuan untuk menghilangkan hubungan linear atau pola statistic antara input dan output, sehingga meningkatkan ketahanan terhadap serangan kriptanalisis.

Dalam fungsi hash, Avalanche Effect menjadi indikator utama menilai kualitas suatu algoritma. Sebuah fungsi hasah memiliki Avalanche Effect yang baik apabila perubahan satu bit pada input menyebabkan setidaknya 50% atau setengah dari bit output berubah secara acak [2].

Secara matematis, Avalanche Effect dapat diukur menggunakan metode Hamming Distance, yaitu jumlah bit yang berbeda antara dua buah output hash. Avalanche Effect dapat dirumuskan sebagai berikut:

$$AE = \frac{Jumlah\ bit\ berbeda}{Panjang\ hash\ (bit)} \times 100\%$$

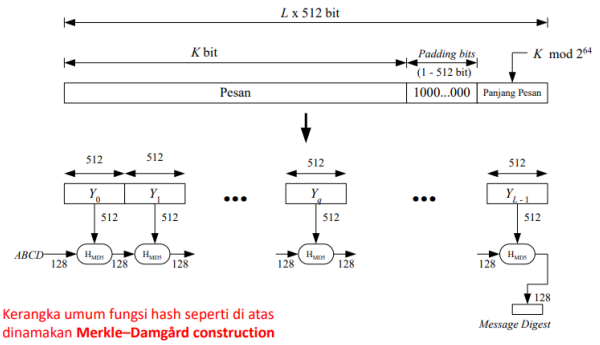
Untuk fungsi hash dengan panjang 256 bit, nilai Avalanche Effect yang ideal berada di 50% atau setara dengan 128 bit yang berubah.

Avalanche Effect berkaitan dengan konsep *confusion and diffusion* yang dikemukakan oleh Claude Shannon. Confusion bertujuan untuk menyembunyikan hubungan statistik yang ada diantara plainteks, cipherteks, dan kunci [3]. Diffusion bertujuan untuk menyebarkan pengaruh setiap bit input ke seluruh bit output sebanyak mungkin [3].

C. SHA-256

SHA-256 merupakan salah satu algoritma fungsi hash kriptografi yang termasuk dalam keluarga SHA-2 (Secure Hash Algorithm 2). SHA-256 dikembangkan oleh NSA (National Security Agency) pada tahun 2001[4].

SHA-256 menghasilkan nilai hash dengan panjang tetap sebesar 256 bit. SHA-256 menggunakan struktur Merkle-Damgård construction [5]. Pesan input dengan panjang sembarang diproses dalam bentuk blok-blok berukuran tetap setelah melalui tahap *padding* [5].



Gambar 1 *Merkle-Damgård Construction* [5]

Pesan dibagi menjadi blok berukuran 512 bit, kemudian setiap blok diproses secara berurutan menggunakan fungsi kompresi hingga menghasilkan nilai hash akhir. Struktur ini memastikan setiap bagian pesan mempengaruhi hasil akhir hash secara kumulatif.

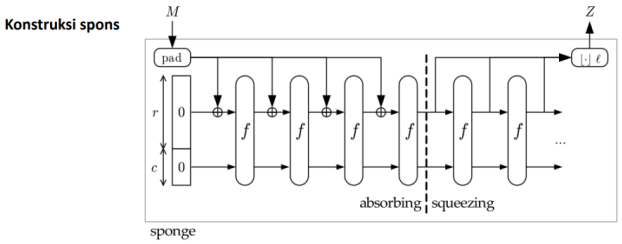
SHA-256 memanfaatkan operasi logika seperti AND, OR, XOR, rotasi bit, dan penjumlahan modulo 2^{32}[5]. Algoritma ini menggunakan delapan variabel kerja awal (*initial hash values*) yang diperoleh dari bagian pecahan akar kuadrat bilangan prima pertama. Pada setiap putaran kompresi, nilai variabel ini diperbarui melalui serangkaian fungsi non-linear untuk menghasilkan difusi dan konfusi yang kuat.

Fungsi kompresi SHA-256 terdiri dari 64 putaran, pada setiap putaran digunakan konstanta tertentu yang berasal dari bagian pecahan akar pangkat tiga bilangan prima [5]. Operasi ini dirancang untuk memastikan bahwa perubahan kecil pada input, seperti perubahan satu bit, akan menyebar ke seluruh bit output hash secara cepat dan acak. Karakteristik ini berkaitan erat dengan penerapan avalanche effect dalam algoritma SHA-256.

D. SHA3-256

SHA3-256 merupakan algoritma fungsi hash kriptografi yang termasuk dalam keluarga SHA-3 (Secure Hash Algorithm 3) yang dikembangkan oleh NSA dan ditetapkan sebagai standar oleh NIST (National Institute of Standards and Technology) pada tahun 2015 [4].

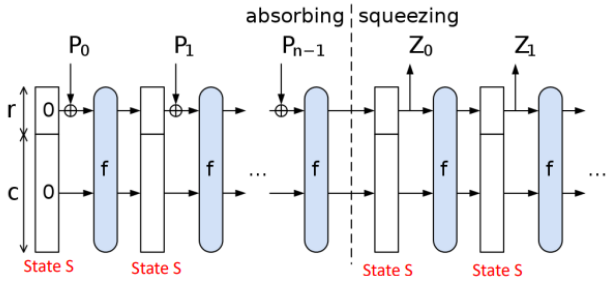
SHA3-256 dikembangkan sebagai standar alternatif dan desain yang berbeda dari keluarga SHA-2. SHA3-256 menggunakan struktur *Sponge Construction* [4].



Gambar 2 *Sponge Construction* [5]

Sponge construction merupakan suatu pendekatan dengan dua tahapan utama yaitu menyerap (*absorb*) dan memeras (*squeeze*) [5].

Pada tahapan *absorbing*, pesan yang telah dipotong-potong menjadi blok berukuran tetap dimasukkan ke dalam bagian *state* melalui operasi logika XOR [5]. Proses ini diikuti dengan pengacakan seluruh isi *state* menggunakan fungsi permutasi kompleks.



Gambar 3 *Absorbing dan Squeezing* [5]

Setelah seluruh pesan diserap, selanjutnya masuk ke tahapan *squeezing*. *Message digest* akan disimpan dalam Z. Kemudian inisialisasi Z dengan string kosong (*null string*). Saat panjang Z belum sama dengan panjang output yang diinginkan, fungsi permutasi akan dijalankan Kembali untuk mengacak *state* sebelum pengambilan bit berikutnya dilakukan.

Kelebihan utama dari struktur ini adalah fleksibilitas dan ketahanannya yang tinggi. *Sponge construction* tahan terhadap *length extension attacks* karena adanya bagian capacity yang tersembunyi dan tidak ditampilkan di output. Selain itu, struktur ini memungkinkan menghasilkan output dengan panjang yang bervariasi sesuai kebutuhan, sehingga tidak hanya sebagai fungsi hash, tetapi juga bisa digunakan dalam *stream cipher*.

III. IMPLEMENTASI

Pada percobaan ini, implementasi dilakukan untuk menganalisis dan membandingkan Avalanche Effect pada fungsi hash SHA-256 dan SHA3-256. Implementasi difokuskan pada pengukuran perubahan bit pada output hash akibat perubahan satu bit pada input. Seluruh proses implementasi dilakukan menggunakan bahasa pemrograman Python karena menyediakan pustaka kriptografi bawaan yang andal serta mendukung analisis numerik dan visualisasi data.

A. Lingkungan dan Alat Implementasi

Implementasi dilakukan menggunakan Python versi 3 dengan memanfaatkan beberapa *library*. *Library* hashlib digunakan untuk menghasilkan nilai hash SHA-256 dan SHA3-256 sesuai standar NIST. Selain itu, *library* numpy digunakan untuk pengolahan data numerik dan perhitungan statistik, sedangkan matplotlib digunakan untuk memvisualisasikan hasil pengujian dalam bentuk grafik.

```
import hashlib
import random
import numpy as np
import matplotlib.pyplot as plt
```

B. Metode Pengujian Avalanche Effect

Metode pengujian avalanche effect dilakukan dengan langkah-langkah sebagai berikut:

1. Sebuah pesan masukan awal dalam bentuk data biner ditentukan sebagai input dasar.
2. Nilai hash dari pesan tersebut dihitung menggunakan algoritma SHA-256 dan SHA3-256.
3. Satu bit pada pesan masukan dimodifikasi secara acak untuk menghasilkan pesan baru yang berbeda minimal dari pesan awal.
4. Nilai hash dari pesan yang telah dimodifikasi kemudian dihitung kembali menggunakan algoritma yang sama.
5. Perbedaan antara hash awal dan hash hasil modifikasi diukur menggunakan Hamming Distance.
6. Nilai Avalanche Effect kemudian dihitung sebagai persentase perubahan bit terhadap total panjang hash, yaitu 256 bit.

```
# 1. Fungsi Hash
def sha256_hash(data: bytes) -> bytes:
      return hashlib.sha256(data).digest()

def sha3_256_hash(data: bytes) -> bytes:
      return
hashlib.sha3_256(data).digest()

# 2. Flip 1 Bit pada Input
def flip_one_bit(data: bytes) -> bytes:
      data = bytearray(data)
      byte_index=random.randint(0,
len(data)-1)
      bit_index = random.randint(0, 7)
      data[byte_index] ^= (1 << bit_index)
      return bytes(data)

# 3. Hamming Distance
def hamming_distance(hash1: bytes, hash2:
bytes) -> int:
      distance = 0
      for b1, b2 in zip(hash1, hash2):
          distance += bin(b1 ^
b2).count('1')
      return distance

# 4. Avalanche Effect (%)
def avalanche_effect(hash1: bytes, hash2:
bytes) -> float:
      hd = hamming_distance(hash1, hash2)
      return (hd / (len(hash1) * 8)) * 100
```

C. Pengujian Berulang dan Pengumpulan Data

Untuk memperoleh hasil yang representatif dan mengurangi bias akibat pemilihan bit tertentu, pengujian dilakukan secara berulang sebanyak 100 kali dan 1000 kali. Pada setiap percobaan, posisi bit yang dimodifikasi dipilih secara acak. Hasil avalanche effect dari setiap percobaan disimpan dan dikumpulkan dalam bentuk data numerik untuk masing-masing algoritma hash.

Data hasil pengujian kemudian dianalisis secara statistik dengan menghitung nilai rata-rata, nilai minimum, nilai maksimum, dan standar deviasi Avalanche Effect. Analisis ini bertujuan untuk mengevaluasi tingkat difusi serta kestabilan Avalanche Effect dari SHA-256 dan SHA3-256.

```
# 5. Pengujian Avalanche
def test_avalanche(hash_func, data: bytes,
trials):
      results = []
      original_hash = hash_func(data)

      for _ in range(trials):
          modified_data =
flip_one_bit(data)
          modified_hash =
hash_func(modified_data)
          ae =
avalanche_effect(original_hash,
modified_hash)
          results.append(ae)

      return results

# 6. Statistik Hasil
def print_statistics(name, data):
      print(f"\n{name}")
      print("-" * 40)
      print(f"Rata-rata
:{np.mean(data):.2f}%")
      print(f"Minimum   :
{np.min(data):.2f}%")
      print(f"Maksimum  :
{np.max(data):.2f}%")
      print(f"Std Dev   :
{np.std(data):.2f}%")
```

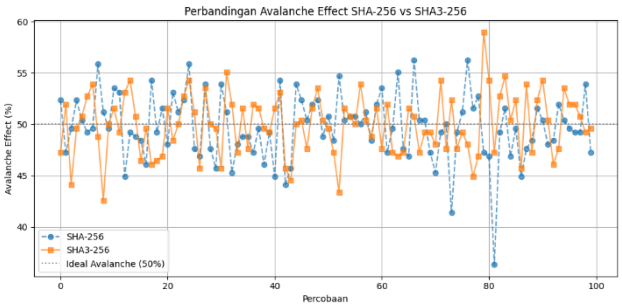
D. Program Utama dan Visualisasi Data

Hasil pengujian Avalanche Effect divisualisasikan dalam bentuk grafik. Grafik ini menampilkan nilai Avalanche Effect untuk setiap percobaan pada kedua fungsi hash, serta garis referensi ideal sebesar 50%. Visualisasi digunakan untuk mempermudah analisis perbandingan pola sebaran, kestabilan, dan kedekatan nilai avalanche effect terhadap kondisi ideal.

```
# 7. Program Utama
if __name__ == "__main__":
  input_data = b"Analisis Avalanche Effect
pada Hash Function"
  trials = 100

  sha256_results =
test_avalanche(sha256_hash, input_data,
trials)
  sha3_results =
test_avalanche(sha3_256_hash, input_data,
trials)

  print_statistics("SHA-256 Avalanche
Effect", sha256_results)
  print_statistics("SHA3-256 Avalanche
Effect", sha3_results)

# 8. Visualisasi
      plt.figure(figsize=(10, 5))
      plt.plot(sha256_results, label="SHA-
256",      marker='o',      linestyle='--',
alpha=0.7)
       plt.plot(sha3_results, label="SHA3-
256",      marker='s',      linestyle='-',
alpha=0.7)
           plt.axhline(50,    color='gray',
linestyle=':',    label='Ideal    Avalanche
(50%)')

        plt.title("Perbandingan  Avalanche
Effect SHA-256 vs SHA3-256")
      plt.xlabel("Percobaan")
      plt.ylabel("Avalanche Effect (%)")
      plt.legend()
      plt.grid(True)
      plt.tight_layout()
      plt.show()
```

IV. HASIL DAN ANALISIS

A. 100 Kali Pengujian



Gambar 4 Grafik Perbandingan Avalanche Effect SHA-256 dan SHA3-256 100 Kali Pengujian

Tabel 1 Avalanche Effect SHA-256 100 Kali Pengujian

SHA-256	
Rata-rata	49.73%
Minimum	36.33%
Maksimum	56.25%
Standar Deviasi	3.20%

Tabel 2 Avalanche Effect SHA3-256 100 Kali Pengujian

SHA3-256	
Rata-rata	49.94%
Minimum	42.58%
Maksimum	58.98%
Standar Deviasi	2.95%

Berdasarkan data hasil 100 kali pengujian di atas, kedua fungsi hash memiliki nilai rata-rata Avalanche Effect mendekati nilai ideal 50%. SHA3-256 menunjukkan nilai rata-rata yang sedikit lebih tinggi dibandingkan SHA-256.

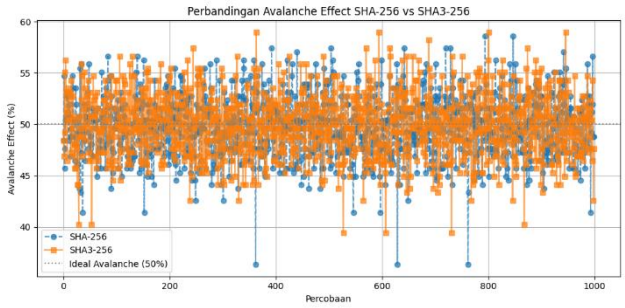
Meskipun perbedaannya relatif kecil, hal ini mengindikasikan bahwa SHA3-256 lebih konsisten dalam menghasilkan perubahan bit output ketika terjadi perubahan minimal pada input. Hal ini juga ditunjukkan pada nilai standar deviasi. SHA-256 memiliki nilai diatas 3%, sedangkan SHA3-256 mampu bertahan di bawah 3%.

Perbedaan paling terlihat pada nilai minimum. SHA-256 memiliki nilai Avalanche Effect paling kecil yaitu 36.33%. Sedangkan SHA3-256 memiliki nilai paling kecil 42.58%. Hal ini menunjukkan bahwa SHA3-256 mampu menghasilkan perbedaan output yang jauh lebih besar daripada SHA-256 dengan margin 6.25%.

Secara umum, untuk 100 kali pengujian, SHA3-256 memberikan hasil Avalanche Effect yang lebih baik dari SHA-256.

B. 1000 Kali Pengujian

Selanjutnya dilakukan 1000 kali pengujian dengan metode yang sama dan didapatkan hasilnya sebagai berikut.



Gambar 5 Grafik Perbandingan Avalanche Effect SHA-256 dan SHA3-256 1000 Kali Pengujian

Tabel 3 Avalanche Effect SHA-256 1000 Kali Pengujian

SHA-256	
Rata-rata	49.73%
Minimum	36.33%
Maksimum	58.59%
Standar Deviasi	3.09%

Tabel 4 Avalanche Effect SHA3-256 1000 Kali Pengujian

SHA3-256	
Rata-rata	49.99%
Minimum	39.45%
Maksimum	58.98%
Standar Deviasi	3.19%

Berdasarkan data hasil 1000 kali pengujian di atas, SHA3-256 memiliki nilai Avalanche Effect yang sangat mendekati 50%. Namun, standar deviasi dari SHA3-256 memiliki nilai yang lebih besar dibandingkan SHA-256 dan nilai standar deviasi pada 100 kali pengujian.

Pada 1000 kali pengujian ini, kedua fungsi hash memiliki nilai Avalanche Effect maksimum yang relatif sama di 58%. Nilai minimum pada SHA3-256 pada pengujian ini juga mengalami penurunan menjadi 39.45%.

C. Analisis Perbandingan

SHA-256 yang berbasis struktur *Merkle-Damgård* memproses pesan secara sekuensial melalui fungsi kompresi yang melibatkan operasi logika dan penjumlahan modulo. Meskipun menghasilkan rata-rata *Avalanche Effect* yang baik (49.73%), nilai minimum yang menyentuh angka 36.33% menunjukkan adanya variasi difusi bit yang cukup lebar pada kondisi tertentu.

Hal ini mengindikasikan bahwa pada beberapa posisi bit input, penyebaran efek perubahan belum merata ke seluruh 256 bit output secara optimal.

Di sisi lain, SHA3-256 yang menggunakan *Sponge Construction* menunjukkan stabilitas yang lebih unggul dengan nilai standar deviasi yang cenderung lebih rendah pada pengujian 100 kali. Tahapan *absorbing* dan *squeezing* yang didukung oleh fungsi permutasi kompleks memungkinkan setiap bit masukan berinteraksi lebih intensif dengan seluruh *state* internal algoritma.

Kemampuan SHA3-256 untuk mempertahankan nilai minimum di angka 39.45%—42.58% (lebih tinggi dari SHA-256) membuktikan bahwa struktur *Sponge* memiliki ketahanan difusi yang lebih konsisten terhadap modifikasi bit tunggal (*1-bit flip*).

Penggunaan *Hamming Distance* sebagai parameter ukur memperlihatkan bahwa kedua algoritma telah berhasil memenuhi kriteria *confusion* dan *diffusion* yang dicanangkan Claude Shannon. Namun, visualisasi grafik menunjukkan bahwa fluktuasi SHA3-256 lebih terkumpul di sekitar garis referensi 50%, menandakan probabilitas perubahan bit yang lebih seragam dan acak.

Hal ini memperkuat SHA3-256 sebagai standar yang tidak hanya menawarkan alternatif desain, tetapi juga keunggulan dalam konsistensi untuk mencegah serangan kriptanalisis.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

- Kedua fungsi hash, baik SHA-256 maupun SHA3-256, telah berhasil memenuhi kriteria *Avalanche Effect* yang baik dengan nilai rata-rata yang sangat mendekati angka ideal 50%
- Dalam pengujian 1000 kali, SHA3-256 menunjukkan performa yang lebih mendekati nilai ideal dengan rata-rata 49.99%, dibandingkan SHA-256 yang berada di angka 49.73%
- SHA3-256 memiliki tingkat konsistensi yang lebih tinggi dalam menyebarkan perubahan bit, ditunjukkan dengan nilai standar deviasi yang lebih rendah (2.95% pada 100 pengujian) dibandingkan SHA-256 (3.20%)
- Perbedaan hasil ini menunjukkan bahwa struktur *Sponge Construction* pada SHA3-256 memberikan distribusi perubahan bit yang lebih stabil dan acak dibandingkan struktur *Merkle-Damgård* pada SHA-256

B. Saran

Untuk pengembangan percobaan serupa di masa mendatang, terdapat beberapa saran yang dapat diterapkan untuk memperluas cakupan analisis.

Pertama, pengujian dapat dilakukan dengan memodifikasi lebih dari satu bit (misalnya 2-bit atau 5-bit *flip*) untuk melihat apakah jumlah bit input yang berubah memiliki korelasi linear terhadap kecepatan pencapaian titik jenuh *Avalanche Effect*.

Kedua, disarankan untuk melakukan pengujian pada berbagai variasi panjang pesan input, mulai dari pesan yang sangat pendek hingga file berukuran besar, untuk mengevaluasi efisiensi fungsi kompresi dan permutasi pada kedua algoritma tersebut.

Terakhir, penelitian selanjutnya dapat menambahkan parameter analisis waktu eksekusi (*computation time*) untuk memberikan perbandingan yang lebih komprehensif antara keamanan kriptografis dan efisiensi performa pada perangkat yang berbeda.

REFERENCES

[1] Munir, Rinaldi. (2025). "Fungsi Hash". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/26-Fungsi-hash-2025.pdf.

[2] https://www.geeksforgeeks.org/computer-networks/avalanche-effect-in-cryptography/, diakses 19.35 WIB, 23 Desember 2025.

[3] Munir, Rinaldi. (2025). "Blok Cipher (Bagian 2)". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/15-Block-Cipher-Bagian2-2025.pdf.

[4] https://www.geeksforgeeks.org/computer-networks/sha-256-and-sha-3/, diakses 20.00 WIB, 23 Desember 2025.

[5] Munir, Rinaldi. (2025). "Beberapa Fungsi Hash". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/27-Beberapa-fungsi-hash-2025.pdf.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

Raga Fatwa Ikhwani
13222001