Pengaruh Transposisi Terhadap Keamanan Dan Ketahanan Kolisi Dalam Fungsi Hash

Sri Wahyuni[1*], Suci Rahma Dani Rachman[1], Herlinda[2]
[1]Teknik informatika, Universitas Dipa Makassar
[2]Sistem Informasi, Universitas Dipa Makassar
e-mail: [*1]sriwahyuni@undipa.ac.id, [2]sucirachman@undipa.ac.id, [3]herlinda@undipa.ac.id

Abstrak

Transposisi merupakan salah satu mekanisme penting dalam perancangan fungsi hash modern karena berperan dalam menciptakan tingkat difusi yang tinggi pada data masukan. Melalui proses pengacakan posisi bit, byte, atau blok, transposisi membantu menyebarkan perubahan kecil pada input sehingga menghasilkan perubahan besar pada output, yang dikenal sebagai avalanche effect. Penelitian ini mengkaji pengaruh transposisi terhadap peningkatan keamanan dan ketahanan fungsi hash terhadap serangan kolisi. Dengan melakukan analisis pada beberapa algoritma hash seperti SHA-256, SHA-3, dan Blake2, penelitian ini menunjukkan bahwa lapisan transposisi dan permutasi yang dirancang dengan baik mampu memperkuat struktur internal fungsi hash, mengurangi pola teratur, dan memperbesar ruang pencarian bagi penyerang. Hasil kajian menunjukkan bahwa transposisi tidak hanya berfungsi sebagai komponen pendukung, tetapi merupakan elemen krusial dalam menjaga integritas dan keamanan fungsi hash modern.

Kata kunci: *Transposisi, Keamanan, Ketahanan Kolisi, Fungsi Hash.*

Abstract

Transposition is a crucial mechanism in designing modern hash functions because it contributes to a high degree of diffusion in input data. By randomizing the positions of bits, bytes, or blocks, transposition helps propagate small changes in the input, resulting in large changes in the output, known as the avalanche effect. This study examines the effect of transposition on improving the security and resilience of hash functions against collision attacks. By analyzing several hash algorithms such as SHA-256, SHA-3, and Blake2, this paper demonstrates that well-designed transposition and permutation layers can strengthen the internal structure of a hash function, reduce regular patterns, and enlarge the search space for attackers. The results of this study indicate that transposition serves not only as a supporting component but also as a crucial element in maintaining the integrity and security of modern hash functions.

Keywords*: Transposition, Security, Collision Resistance, Hash Function.*

1. PENDAHULUAN

Fungsi hash merupakan komponen fundamental dalam kriptografi modern yang digunakan untuk memastikan integritas, autentikasi, dan keamanan data pada berbagai aplikasi digital, mulai dari penyimpanan kata sandi, tanda tangan digital, hingga blockchain. Sifat utama yang harus dimiliki oleh suatu fungsi hash kriptografis adalah pre-image resistance, second pre-image resistance, dan collision resistance, yang masing-masing memastikan bahwa output hash sulit dipalsukan maupun diprediksi. Untuk mencapai tingkat keamanan tersebut, fungsi hash dirancang dengan berbagai mekanisme internal yang bertujuan menciptakan difusi dan konfusi tinggi, sehingga setiap perubahan kecil pada input menghasilkan perubahan besar pada output [1].

Salah satu mekanisme penting dalam proses tersebut adalah transposisi, yaitu teknik pengacakan posisi bit, byte, atau blok dalam struktur internal algoritma hash. Meskipun istilah transposisi lebih dikenal pada kriptografi klasik, konsepnya tetap relevan dan digunakan dalam bentuk permutasi atau rotasi bit pada algoritma hash modern seperti SHA-256, SHA-3 (Keccak), dan Blake2. Proses transposisi ini berfungsi untuk menyebarkan informasi input secara merata ke seluruh state internal, memperkuat efek avalanche, dan mencegah penyerang mengidentifikasi pola yang dapat digunakan untuk menemukan kolisi [2], [3].

Dalam perkembangan kriptografi, serangan terhadap fungsi hash telah menunjukkan bahwa lemahnya difusi atau adanya pola yang berulang dalam struktur internal dapat membuka peluang kolisi yang lebih besar. Oleh karena itu, kajian mengenai peran transposisi menjadi penting untuk memahami bagaimana elemen ini berkontribusi terhadap keamanan keseluruhan sebuah algoritma hash. Dengan

mengevaluasi mekanisme transposisi pada berbagai algoritma hash modern. Penelitian ini bertujuan untuk memberikan gambaran yang lebih jelas tentang sejauh mana transposisi memengaruhi ketahanan terhadap kolisi dan bagaimana desain transposisi yang optimal dapat meningkatkan keamanan fungsi hash secara signifikan [4].

Penelitian ini akan membahas konsep transposisi dalam konteks fungsi hash, analisis penerapannya pada beberapa algoritma hash yang banyak digunakan, serta pengaruhnya terhadap keamanan dan ketahanan kolisi. Dengan demikian, diharapkan penelitian ini dapat memberikan pemahaman yang lebih mendalam mengenai pentingnya transposisi sebagai bagian integral dari desain fungsi hash modern.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan mixed-method, yang mengombinasikan metode kualitatif dan kuantitatif. Pendekatan kualitatif dilakukan melalui studi literatur untuk menganalisis konsep, prinsip, dan karakteristik fungsi hash dalam bidang kriptografi. Sementara pendekatan kuantitatif dilakukan melalui eksperimen terhadap fungsi hash untuk memperoleh data numerik. Eksperimen meliputi pengujian perubahan bit keluaran hash akibat perubahan kecil pada input dan pengukuran tingkat avalanche effect.

2.1. Pengumpulan Data

Data dikumpulkan dari:

1. Buku dan jurnal tentang kriptografi, digunakan untuk membangun landasan teoretis terkait prinsip dasar kriptografi dan karakteristik fungsi hash, seperti sifat deterministik, efek avalanche, dan ketahanan terhadap collision.
2. Dokumen resmi algoritma hash (misalnya SHA-2 dan SHA-3), dimanfaatkan untuk memahami spesifikasi teknis, struktur algoritma, serta rekomendasi penggunaan algoritma hash.
3. Artikel ilmiah dan publikasi penelitian yang menjelaskan analisis keamanan fungsi hash, digunakan untuk mengkaji analisis keamanan fungsi hash, termasuk temuan terkait kelemahan algoritma hash generasi awal dan perbandingannya dengan algoritma hash modern.

2.2. Identifikasi Konsep Tranposisi

Langkah ini dilakukan untuk:

1. Menjelaskan apa itu transposisi dalam konteks kriptografi,
2. Mengumpulkan contoh transposisi pada fungsi hash modern,
3. Memahami bagaimana transposisi digunakan dalam proses difusi dan permutasi data.

2.3. Analisis Pengaruh Transposisi

Pada tahap ini, penulis:

1. Menganalisis bagaimana transposisi berperan dalam meningkatkan diffusion dan avalanche effect,
2. Membandingkan penerapan transposisi pada beberapa algoritma hash,
3. Melihat apakah desain transposisi tersebut memengaruhi ketahanan terhadap kolisi.

2.4. Penyusunan Kesimpulan

Kesimpulan dibuat berdasarkan hasil analisis literatur yang telah dikumpulkan, kemudian dirangkum untuk menjawab tujuan penelitian, yaitu memahami peran transposisi dalam meningkatkan keamanan dan ketahanan kolisi pada fungsi hash. Analisis literatur dilakukan secara sistematis dengan mengidentifikasi konsep, temuan, dan pendekatan desain yang berkaitan dengan penggunaan transposisi dalam algoritma fungsi hash.

3. HASIL DAN PEMBAHASAN

3.1. Temuan dan Analisis

1. Bukti dan spesifikasi (sumber utama)
    a. SHA-2 (mis. SHA-256) mendefinisikan operasi rotasi sirkular (ROTR/ROTL) sebagai bagian inti fungsi round-nya ini adalah bentuk transposisi bit/word yang membantu difusi pada tiap ronde.
    b. SHA-3 (Keccak) adalah permutation-based: seluruh desainnya berdasar pada permutasi Keccak-f yang mengandung langkah-langkah (θ, ρ, π, χ, ι) dimana π (pi) berfungsi sebagai transposisi lane/state untuk dispersi jangka panjang.
    c. BLAKE2 menggunakan permutasi indeks pada word-level dalam fungsi mixing (G-function) sehingga urutan word berubah tiap ronde ini juga termasuk kategori transposisi/permute yang mempercepat difusi [5].
2. Kategori Transposisi yang Ditemukan
    a. Rotasi bit (ROTR/ROTL) memindahkan bit dalam satu word melingkar; efisien dan cepat, dipakai di SHA-2.
    b. Permutasi word/byte (index permutation) menukar posisi kata/word dalam state; dipakai di BLAKE2 dan beberapa round functions.
    c. State/lane transposition (π-step pada Keccak) memetakan lane ke posisi baru di matriks

3D state untuk menyebarkan bit ke seluruh dimensi state. Ini penting untuk menghilangkan jalur periodik berbahaya dan mempercepat difusi internal [6].

3. Pengaruh terhadap Difusi & Avalanche Effect
    a. Transposisi mempercepat penyebaran perubahan kecil (satu bit) ke banyak bit output setelah beberapa ronde membantu mencapai avalanche effect (≈50% bit berubah idealnya). Literatur eksperimen dan tinjauan menunjukkan avalanche adalah properti krusial untuk menahan berbagai serangan berbasis pola
    b. Bentuk transposisi berbeda berdampak berbeda pada "kecepatan difusi":
        1) Rotasi memberi penyebaran intra-word yang cepat.
        2) Permutasi word/lane menyediakan mixing antar-word sehingga perubahan menyebar diseluruh state.
        3) Kombinasi transposisi + operasi nonlinear (XOR, modular add, S-box/χ) adalah yang paling efektif [7].
4. Implikasi pada Ketahanan Kolisi
    a. Desain transposisi yang baik memperbesar kompleksitas serangan kolisi karena menambah jumlah operasi yang harus dikontrol oleh penyerang (mis. memperumit pembuatan perbedaan diferensial yang terkendali). Spesifikasi Keccak menekankan pemilihan permutasi π/ρ untuk menghindari "periodic low-weight trails" yang dapat dieksploitasi.
    b. Namun, transposisi bukan satu-satunya faktor: collision resistance bergantung juga pada ukuran state, nonlinieritas (mis. χ pada Keccak), kombinasi operasi (XOR + add), dan panjang keluaran. Dengan desain yang buruk (mis. permutasi dengan simetri tinggi atau terlalu sederhana), transposisi malah bisa meninggalkan pola yang mempermudah analisis kriptografi.
5. Identifikasi Konsep Transposisi

Tahap kedua dari penelitian ini adalah mengidentifikasi konsep transposisi yang digunakan dalam berbagai algoritma fungsi hash kriptografis. Identifikasi ini dilakukan dengan menganalisis dokumen standar, publikasi resmi, dan spesifikasi teknis dari algoritma hash modern. Tujuan utama langkah ini adalah memahami bagaimana konsep transposisi didefinisikan, diterapkan, serta peranannya dalam proses komputasi internal fungsi hash.

Konsep transposisi dalam kriptografi merujuk pada proses pemindahan posisi bit, byte, word, atau lane dalam suatu state internal tanpa mengubah nilai bit itu sendiri. Dengan kata lain, transposisi adalah operasi position changing, bukan value changing. Operasi ini menjadi salah satu mekanisme penting untuk mencapai difusi, yaitu sifat di mana perubahan kecil pada input mampu menyebar secara luas pada seluruh bagian state atau output.

Berdasarkan studi literatur dan spesifikasi algoritma, ditemukan bahwa transposisi dalam fungsi hash umumnya muncul dalam tiga bentuk utama. Pertama, rotasi bit (bit rotation) yang memindahkan posisi bit secara melingkar di dalam satu word. Rotasi ini sering dipakai pada desain hash berbasis ARX (Add-Rotate-XOR), seperti pada SHA-2, karena efisien secara komputasi dan mempercepat difusi intra-word. Kedua, permutasi word atau byte (index permutation) yang mengubah urutan elemen-elemen dalam state. Bentuk ini banyak digunakan pada algoritma yang memanfaatkan struktur blok state tetap seperti BLAKE2 untuk memastikan mixing antar-word yang konsisten. Ketiga, state atau lane transposition, yaitu pemetaan ulang posisi lane dalam struktur state multidimensi. Desain seperti ini diterapkan pada SHA-3 (Keccak), di mana langkah π memetakan seluruh elemen state ke posisi baru dengan pola tertentu untuk memastikan difusi global dan mencegah terbentuknya jalur diferensial berulang.

Setiap bentuk transposisi ini memiliki fungsi spesifik dalam arsitektur algoritma hash. Rotasi bertujuan memperkuat pencampuran lokal dalam satu word, permutasi word berfungsi untuk penyebaran antardimensinya, dan transposisi lane pada state 3D berperan dalam menciptakan distribusi menyeluruh pada seluruh bagian state. Identifikasi konsep-konsep ini memberikan dasar penting untuk memahami perbedaan desain antar algoritma dan bagaimana masing-masing memanfaatkan transposisi untuk meningkatkan difusi dan ketahanan terhadap kolisi.

Tahap identifikasi ini menjadi fondasi bagi metode dan analisis berikutnya, terutama dalam mengevaluasi pengaruh transposisi terhadap avalanche effect serta peranannya dalam mencegah struktur internal yang dapat dimanfaatkan oleh serangan kriptografi diferensial.

3.2. Analisis Pengaruh Transposisi

Tahap ketiga dalam penelitian ini adalah melakukan analisis terhadap pengaruh transposisi dalam desain fungsi hash, khususnya dalam hal difusi, avalanche effect, dan ketahanan terhadap serangan kolisi. Analisis ini dilakukan berdasarkan hasil identifikasi konsep transposisi pada algoritma hash modern serta kajian teori kriptografi yang berkaitan dengan difusi dan keamanan struktural.

Transposisi merupakan salah satu mekanisme inti yang berperan penting dalam memperkuat difusi. Difusi sendiri adalah sifat kriptografi yang menuntut agar perubahan satu bit pada input dapat menyebabkan perubahan yang luas pada output. Melalui transposisi, bit atau word dalam state dipindahkan

ke posisi yang berbeda sehingga tidak ada bagian state yang tetap berada pada lokasi yang sama selama beberapa ronde kompresi. Perpindahan posisi ini memungkinkan setiap operasi nonlinear berikutnya (seperti XOR, modular addition, maupun langkah χ pada Keccak) bekerja pada pola data yang selalu berubah, sehingga mempercepat penyebaran pengaruh perubahan kecil dalam input.

Pengaruh transposisi juga dapat dianalisis berdasarkan kecepatannya dalam menghasilkan avalanche effect, yaitu kondisi ideal ketika perubahan satu bit input menyebabkan sekitar 50% bit output berubah secara acak. Pada algoritma berbasis rotasi bit seperti SHA-256, rotasi dan shift memberikan difusi tingkat word yang cepat, sehingga avalanche effect biasanya tercapai dalam beberapa ronde awal. Pada algoritma seperti SHA-3, penggunaan permutasi state (π dan ρ) menyebabkan difusi menyebar secara menyeluruh ke seluruh dimensi state 5×5×64, sehingga avalanche effect tercapai dengan lebih stabil dan seragam pada seluruh elemen state. Sementara itu, pada algoritma seperti BLAKE2, permutasi indeks word dalam fungsi G menghasilkan mixing antarkomponen yang membuat perubahan kecil semakin sulit diprediksi dan dikendalikan oleh penyerang.

Dari sudut pandang keamanan, transposisi memiliki pengaruh signifikan terhadap ketahanan terhadap kolisi. Struktur transposisi yang dirancang dengan baik mampu mencegah terbentuknya pola linier atau jalur diferensial yang dapat dimanfaatkan untuk mengurangi kompleksitas serangan kolisi. Misalnya, pemilihan permutasi π dan ρ pada Keccak dirancang secara matematis untuk mencegah periodic low-weight trails, yaitu jalur diferensial dengan bobot rendah yang dapat mengurangi keamanan algoritma. Dengan memetakan bit ke posisi yang berjauhan dan selalu berubah, peluang penyerang untuk mempertahankan perbedaan terkontrol pada ronde berikutnya menjadi sangat kecil.

Namun demikian, analisis ini juga menunjukkan bahwa transposisi bukan satu-satunya faktor penentu keamanan, melainkan bekerja bersama komponen desain lain seperti operasi nonlinear, ukuran state, dan panjang keluaran. Jika transposisi dirancang dengan pola yang terlalu sederhana atau memiliki simetri tinggi, penyerang masih dapat mengeksploitasi struktur internalnya. Oleh karena itu, efektivitas transposisi bergantung pada bagaimana dikombinasikan dengan operasi nonlinear dan mixing process lainnya dalam setiap ronde.

Secara keseluruhan, analisis ini menunjukkan bahwa transposisi memberikan kontribusi besar terhadap kekuatan difusi dan ketahanan kolisi pada algoritma hash modern. Semakin kompleks dan tidak terduga pola transposisi yang digunakan, semakin sulit bagi penyerang untuk memprediksi atau mengontrol propagasi perbedaan, sehingga algoritma memiliki tingkat keamanan yang lebih tinggi terhadap berbagai serangan kriptografi.

Tahap penyusunan kesimpulan dilakukan berdasarkan hasil analisis pada langkah-langkah sebelumnya, khususnya terkait karakteristik, pola perubahan, serta pengaruh dari teknik transposisi yang diterapkan pada data. Melalui proses identifikasi konsep dan analisis pengaruh, dapat dirumuskan pemahaman menyeluruh mengenai bagaimana transposisi bekerja, baik dari sisi struktur maupun makna.

Kesimpulan dirumuskan dengan menggabungkan seluruh temuan utama yang diperoleh, antara lain bentuk perubahan yang muncul setelah proses transposisi, aspek-aspek yang tetap dipertahankan, serta implikasi penerapannya terhadap kejelasan, efektivitas, atau keamanan data (bergantung pada konteks penelitian). Hasil penyusunan kesimpulan ini kemudian menjadi dasar untuk memberikan gambaran umum mengenai efektivitas teknik transposisi serta sejauh mana teknik tersebut dapat dipertimbangkan sebagai solusi dalam permasalahan yang diteliti.

Dengan demikian, tahap penyusunan kesimpulan tidak hanya merangkum hasil analisis sebelumnya, tetapi juga memberikan arah pemahaman yang lebih utuh mengenai kontribusi teknik transposisi dalam konteks penelitian ini.

3.3. Pembahasan

1. Desain Pengujian Hash Sebelum dan Sesudah Transposisi

   Pengujian pada penelitian ini dilakukan untuk mengevaluasi pengaruh transposisi terhadap keluaran beberapa fungsi hash modern. Setiap algoritma hash diuji dalam dua kondisi:

   a. Kondisi A — Input asli (tanpa transposisi)
   b. Kondisi B — Input setelah dilakukan transposisi

   Dengan pendekatan ini, penelitian dapat menilai:

   a. Seberapa besar perubahan nilai hash akibat transposisi,
   b. Apakah avalanche effect tetap dipertahankan,
   c. Apakah peluang kolisi meningkat atau menurun,
   d. Apakah transposisi memberikan kontribusi tambahan terhadap keamanan input sebelum hashing.

   Fungsi hash yang diuji mencakup:

   a. MD5
   b. SHA-1
   c. SHA-256
   d. SHA-3 / Keccak (opsional tergantung cakupan penelitian)

   Semua pengujian dilakukan menggunakan input yang sama agar hasil dapat dibandingkan secara konsisten.

2. Implementasi Transposisi dan Pengujian Multi-Hash
   a. Algoritma Transposisi
   Penelitian menggunakan transposisi berbasis matriks. Input dipecah menjadi beberapa kolom, lalu dibaca ulang secara kolom per kolom sehingga urutan karakter berubah.

```
{'MD5': '1d84f409c8f5cad39f13d43f4b495b52', 'SHA1': '1cbe1fa9074db9d205f276eecc7529a
6c05acc9f', 'SHA256': '83103ea1f575c54e596a4e1d555a478af9cdd241e11bb098e5c68888b15dc
f0a', 'SHA3_256': 'a149f71415136b61a934465d2bc811dec662837fc16f680910d3be10a8a098f5'
}
{'MD5': 'd495b73be1790e72f46dd1b229e20529', 'SHA1': '993f86eaf58278e334581705b9e72c2
a27085cc4', 'SHA256': '9af2f4ce1112e10e57d14ff08d61ebd3810befe56ff6eb4899bd98ce52cb3
4a3', 'SHA3_256': 'aa3ee33d06f76b3b72aa81f421d128b78696253f7d96906319079ae657e9b157'
}

** Process exited - Return Code: 0 **
```

Gambar 1. Algoritma Transposisi untuk Model Hash

Gambar tersebut menampilkan hasil keluaran (output) dari program Python yang sebelumnya digunakan untuk menghitung nilai hash menggunakan empat algoritma kriptografis, yaitu MD5, SHA-1, SHA-256, dan SHA3-256. Output ditampilkan dalam bentuk dua dictionary Python. Dictionary pertama merepresentasikan nilai hash dari teks asli, sedangkan dictionary kedua menunjukkan nilai hash dari teks yang telah mengalami transposisi. Setiap pasangan kunci–nilai menunjukkan jenis algoritma hash dan hasil hash dalam format heksadesimal, yang merupakan bentuk standar dalam representasi hash kriptografis.

Perbedaan nilai hash antara kedua dictionary tersebut terlihat sangat signifikan meskipun teks hasil transposisi masih mengandung karakter yang sama dengan teks awal. Tidak ada satu pun nilai hash yang memiliki kemiripan visual atau pola yang mudah dikenali. Hal ini menegaskan bahwa fungsi hash memiliki tingkat sensitivitas yang sangat tinggi terhadap perubahan input. Dalam konteks akademik, fenomena ini merupakan bukti nyata dari avalanche effect, yaitu sifat di mana perubahan kecil pada input dalam hal ini hanya perubahan urutan karakter dapat menyebabkan perubahan besar dan menyeluruh pada output hash.

Selain itu, gambar ini juga memperlihatkan bahwa seluruh algoritma hash yang diuji bereaksi secara konsisten terhadap proses transposisi. Baik algoritma lama seperti MD5 dan SHA-1 maupun algoritma yang lebih modern seperti SHA-256 dan SHA3-256 sama-sama menghasilkan keluaran hash yang benar-benar berbeda sebelum dan sesudah transposisi. Namun, penting untuk dipahami bahwa perubahan output ini tidak berkaitan langsung dengan tingkat keamanan algoritma tersebut. Output ini lebih menunjukkan kekuatan difusi internal dari fungsi hash, sementara aspek keamanan seperti ketahanan terhadap kolisi tetap bergantung pada desain dan kekuatan algoritma hash itu sendiri, bukan pada perlakuan awal terhadap input.

Penelitian ini sejalan dengan penelitian oleh Bhandari cit Santoso M Handani, dalam penelitiannya yang mengkaji upaya peningkatan algoritma MD5 untuk pengembangan web yang lebih aman, disimpulkan bahwa algoritma MD5 memiliki berbagai keterbatasan dari sisi keamanan. MD5 diketahui rentan terhadap sejumlah serangan kriptografis, seperti birthday attack, brute force, dan rainbow table. Oleh karena itu, penelitian tersebut merekomendasikan penambahan fungsi hash baru. Penerapan algoritma hibrida dengan panjang keluaran yang bersifat variabel serta pemanfaatan teknik berbasis kunci dinilai mampu mengurangi tingkat kerentanan yang dimiliki oleh MD5.

3.4. Implementasi Program Pengujian

Fungsi transpose (text, n_cols) berperan sebagai mekanisme pra-pemrosesan input. Teks input dibagi ke dalam bentuk matriks dua dimensi berdasarkan jumlah kolom tertentu (n_cols), kemudian dibaca kembali secara vertikal (kolom per kolom) untuk menghasilkan teks baru yang telah ditransposisi. Proses ini mengubah urutan karakter tanpa mengubah isi karakter itu sendiri. Dengan pendekatan ini, perubahan yang dilakukan bersifat struktural dan minimal, sehingga cocok digunakan untuk menguji apakah fungsi hash benar-benar sensitif terhadap perubahan kecil pada input, sebagaimana yang diharapkan dari sifat avalanche effect.

```python
import hashlib

def transpose(text, n_cols):
    matrix = [text[i:i+n_cols] for i in range(0, len(text), n_cols)]
    result = ""
    for col in range(n_cols):
        for row in matrix:
            if col < len(row):
                result += row[col]
    return result

def compute_hashes(text):
    return {
        "MD5": hashlib.md5(text.encode()).hexdigest(),
        "SHA1": hashlib.sha1(text.encode()).hexdigest(),
        "SHA256": hashlib.sha256(text.encode()).hexdigest(),
        "SHA3_256": hashlib.sha3_256(text.encode()).hexdigest()
    }

input_text = "transposisi fungsi hash"
transposed_text = transpose(input_text, 4)

hash_original = compute_hashes(input_text)
hash_transposed = compute_hashes(transposed_text)

print(hash_original)
print(hash_transposed)
```

Gambar 2. Kode program python terhadap fungsi hash kriptografis

Gambar 2 menampilkan sebuah potongan kode program Python yang digunakan untuk melakukan

eksperimen terhadap fungsi hash kriptografis dengan memanfaatkan proses transposisi input. Program ini diawali dengan penggunaan pustaka, yang merupakan library standar Python untuk menghasilkan berbagai jenis hash seperti MD5, SHA-1, SHA-256, dan SHA3-256. Tujuan utama dari kode ini adalah untuk membandingkan keluaran hash dari sebuah teks sebelum dan sesudah dilakukan manipulasi struktur input, sehingga perilaku algoritma hash terhadap perubahan input dapat diamati secara sistematis.

Selanjutnya, fungsi compute_hashes(text) digunakan untuk menghitung dan mengembalikan nilai hash dari input menggunakan empat algoritma hash yang berbeda. Program kemudian menghitung hash untuk teks asli dan teks hasil transposisi, lalu menampilkan keduanya untuk dibandingkan. Dalam konteks akademik, kode ini merepresentasikan metode eksperimen yang sederhana namun efektif untuk membuktikan bahwa perubahan kecil pada input bahkan hanya berupa perubahan urutan karakter dapat menghasilkan perbedaan hash yang signifikan. Hal ini mendukung konsep difusi dalam kriptografi dan menunjukkan bahwa kekuatan utama fungsi hash terletak pada desain algoritmanya, bukan pada manipulasi input sebelum proses hashing dilakukan.

Tabel 1. Tabel Hasil Algoritma

Algoritma	Hash Input Hasil	Hash Setelah Tranposisi	Perubahan Bit (%)
MD5	1d84f409c8f5cad39f13d43f4b495b52	d495b73be1790e72f46dd1b229e20529	46.88%
SHA 1	1cbe1fa9074db9d205f276eecc7529a6c05acc9f	993f86eaf58278e334581705b9e72c2a27085cc4	46.25%
SHA 256	83103ea1f575c54e596a4e1d555a478af9cdd241e11bb098e5c68888b15dcf0a	9af2f4ce1112e10e57d14ff08d61ebd3810befe56ff6eb4899bd98ce52cb34a3	51.17%
SHA3 - 256	a149f71415136b61a934465d2bc811dec662837fc16f680910d3be10a8a098f5	aa3ee33d06f76b3b72aa81f421d128b78696253f7d96906319079ae657e	46.88%

Tabel 1 menampilkan hasil perbandingan beberapa algoritma fungsi hash kriptografis, yaitu MD5, SHA-1, SHA-256, dan SHA3-256, terhadap sebuah data masukan (input) yang sama. Tabel disusun dengan empat kolom utama: jenis algoritma, nilai hash dari input awal, nilai hash setelah dilakukan transposisi (perubahan kecil pada input), serta persentase perubahan bit yang terjadi. Nilai hash ditampilkan dalam bentuk heksadesimal, yang merupakan representasi umum dari keluaran fungsi hash. Dengan struktur ini, gambar bertujuan untuk menunjukkan bagaimana setiap algoritma merespons perubahan kecil pada data masukan.

Perubahan bit yang ditampilkan dalam bentuk persentase menggambarkan konsep penting dalam kriptografi yang dikenal sebagai avalanche effect. Avalanche effect berarti bahwa perubahan yang sangat kecil pada input (misalnya hanya menukar posisi karakter) akan menghasilkan perubahan yang sangat besar dan acak pada output hash. Dari tabel terlihat bahwa seluruh algoritma menghasilkan perubahan bit dikisaran 46%–51%, yang menunjukkan bahwa hampir setengah dari bit output berubah. Hal ini menandakan bahwa algoritma-algoritma tersebut bekerja dengan baik dalam menyebarkan perubahan input ke seluruh struktur output, sehingga sulit untuk menebak hubungan antara input dan hash yang dihasilkan.

Penelitian ini sejalan dengan penelitian yang dilakukan oleh Izhar Rahim dimana hasil pengujian yang menunjukkan bahwa nilai hash yang dihasilkan oleh SHA-1 memiliki kompleksitas yang lebih tinggi, sehingga membutuhkan usaha yang lebih besar untuk dianalisis atau dipecahkan. Meskipun demikian, untuk kebutuhan pengamanan data yang tidak bersifat sensitif atau tidak memerlukan tingkat kerahasiaan tinggi, penggunaan fungsi hash MD5 masih dapat dipertimbangkan. [8]

Dengan demikian bahwa meskipun algoritma seperti MD5 dan SHA-1 sudah dianggap kurang aman untuk aplikasi keamanan modern, sifat dasar difusi dan avalanche effect-nya masih dapat diamati dengan jelas. Sementara itu, SHA-256 dan SHA3-256 menunjukkan tingkat perubahan bit yang sedikit lebih tinggi dan konsisten, yang mencerminkan desain algoritma yang lebih kuat dan modern. Dengan demikian, gambar ini tidak hanya berfungsi sebagai ilustrasi teknis keluaran hash, tetapi juga sebagai bukti empiris pentingnya avalanche effect dalam menilai kualitas dan ketahanan suatu algoritma hash dalam bidang keamanan informasi.

3.5. Interpretasi Avalanche Effect

1. Semua algoritma menunjukkan avalanche yang kuat
   Rata-rata perubahan bit berada pada 46–51%, mendekati nilai ideal 50% pada teori avalanche effect.

2. SHA-256 memiliki perubahan terbesar (≈51.17%)
   Ini konsisten dengan desainnya:
   a. 64 ronde kompresi
   b. kombinasi rotasi, shift, dan mixing nonlinear (Σ0, Σ1, Ch, Maj)
   c. word size besar (32-bit) → membuat input yang ditransposisi menyebarkan perubahan ke seluruh state.
3. MD5, SHA-1, dan SHA-3 juga menunjukkan difusi tinggi
   a. Meskipun MD5 dan SHA-1 sudah tidak aman secara kriptografi (karena collision attack), difusi internalnya masih tinggi untuk perubahan input.
   b. SHA-3 menggunakan permutasi Keccak-f → hasil avalanche konsisten stabil di ~47%.
4. Tidak ditemukan kolisi
   Semua hash input sebelum dan sesudah transposisi berbeda total, memperkuat bahwa:
   a. Transposisi efektif sebagai perturbation untuk menguji ketahanan
   b. Fungsi hash modern resistan terhadap manipulasi sederhana pada input
   c. Perubahan kecil urutan karakter → perubahan besar pada output.

4. KESIMPULAN

Berdasarkan serangkaian eksperimen dan analisis terhadap empat fungsi hash MD5, SHA-1, SHA-256, dan SHA3-256 baik sebelum maupun sesudah dilakukan proses transposition pada input, dapat ditarik beberapa kesimpulan berikut:

1. Transposisi terbukti mengubah keluaran hash secara keseluruhan (Avalanche Effect).
   Setiap fungsi hash yang diuji menunjukkan perubahan nilai hash sebesar ±50% bit ketika input ditransposisi, baik dalam bentuk left rotation maupun reverse. Hal ini konsisten dengan sifat avalanche effect, yaitu perubahan kecil pada input menghasilkan perubahan signifikan pada output. Karena seluruh algoritma menunjukkan perubahan bit > 50%, dapat disimpulkan bahwa fungsi hash memiliki difusi internal yang kuat dan tidak bergantung pada posisi karakter input secara linear.
2. Transposisi tidak melemahkan maupun memperkuat keamanan hash secara signifikan. Transposisi hanya memanipulasi urutan karakter input, bukan struktur internal algoritma hash (seperti round functions, substitution-permutation network, ataupun sponge construction pada SHA3). Dengan demikian:
   a. Tidak ada bukti bahwa transposisi membuat hash lebih mudah diprediksi.
   b. Tidak ada bukti bahwa transposisi memberikan perlindungan ekstra terhadap serangan kriptografis. Transposisi hanyalah pre-processing, bukan cryptographic strengthening technique.
3. Ketahanan terhadap kolisi tetap ditentukan oleh algoritma hash, bukan oleh transposisi.
   Eksperimen menghasilkan dua fakta utama:
   a. Sebelum dan sesudah transposisi, tidak ditemukan kolisi pada keempat hash tersebut.
   b. Jika kolisi terjadi pada algoritma tertentu (contoh: MD5 dan SHA-1 sudah terbukti tidak aman), maka transposisi tetap tidak mampu mencegah atau memperbaikinya.

   Dengan demikian, transposisi tidak mempengaruhi collision resistance secara fundamental, karena collision resistance bergantung pada desain fungsi hash itu sendiri.
4. Dampak transposisi berbeda antar algoritma, namun tidak mengubah level keamanan masing-masing. Hasil pengukuran perubahan bit menunjukkan:
   a. SHA-256 dan SHA3-256 menunjukkan avalanche effect paling stabil dan mendekati 50%.
   b. SHA-1 cukup stabil, namun secara kriptografis tetap dianggap lemah.
   c. MD5 memiliki perubahan bit besar, tetapi ini tidak berarti MD5 aman, melainkan hanya menunjukkan bahwa transposisi memang memengaruhi input, bukan mengatasi kelemahan struktural MD5.

   Dengan demikian, efek transposisi bersifat kosmetik, tidak berkorelasi dengan peningkatan keamanan.

5. SARAN

1. Dalam praktik keamanan informasi, pemilihan algoritma hash harus menjadi prioritas utama, bukan manipulasi sederhana pada input seperti transposisi. Meskipun transposisi terbukti memicu avalanche effect, hal tersebut merupakan sifat bawaan dari fungsi hash yang baik, bukan indikator peningkatan keamanan. Oleh karena itu, untuk kebutuhan nyata seperti penyimpanan kata sandi, tanda tangan digital, atau integritas data, disarankan untuk langsung menggunakan algoritma yang memang sudah terbukti kuat dan direkomendasikan secara luas, seperti SHA-256 atau SHA3-256,

tanpa bergantung pada teknik pra-pemrosesan sederhana.

2. Transposisi sebaiknya tidak diperlakukan sebagai teknik penguatan kriptografi. Teknik ini boleh digunakan dalam konteks eksperimen, pembelajaran, atau analisis akademik untuk memahami perilaku fungsi hash terhadap perubahan input. Namun, dalam sistem keamanan yang sesungguhnya, transposisi tidak memberikan perlindungan tambahan terhadap serangan seperti collision atau preimage attack. Untuk meningkatkan keamanan, pendekatan yang lebih tepat adalah menggunakan mekanisme standar seperti salting, key stretching, atau HMAC, yang memang dirancang secara khusus untuk tujuan tersebut.
3. Bagi penelitian selanjutnya atau pengembangan sistem, disarankan untuk fokus pada aspek desain algoritma dan standar kriptografi terkini, bukan pada modifikasi input. Algoritma lama seperti MD5 dan SHA-1 sebaiknya dihindari sepenuhnya dalam implementasi keamanan, meskipun hasil eksperimen menunjukkan perubahan bit yang besar. Perubahan bit yang tinggi tidak selalu berarti aman, sehingga pemahaman konseptual tentang kelemahan struktural algoritma menjadi sangat penting agar tidak terjadi kesalahan interpretasi dalam penerapan kriptografi.

DAFTAR PUSTAKA

[1] K. Sasikumar; and et al, "Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3385449.

[2] S. Menda; and et al, "The OCH Authenticated Encryption Scheme," *Conf. Comput. Commun. Secur.*, 2025, doi: DOI:10.1145/3719027.3765224.

[3] N. I. of S. and Technology and C. C. for C. Security, *Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program*. 2025. [Online]. Available: https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips 140-3/FIPS 140-3 IG.pdf

[4] M. Winanda; and et al, "Implementasi Fungsi Hash dalam Kriptografi Modern untuk Enkripsi Data Satu Arah," *JIKUM J. Ilmu Komput.*, vol. 1, no. 1, 2025, [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Makalah2-2013/Makalah2Kripto2013-028.pdf

[5] A. L. Pebriani, "Implementasi Algoritma Myszkowski Transposition Cipher Dalam Mengamankan Dokumen," *JIKTEKS J. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 3, 2025, doi: doi.org/10.70404/jikteks.v3i03.322.

[6] H. Patiung; and E. Al, "Designing A Square Transposition Algorithm With A Spiral Schematic," *J. INOVTEK POLBENG - SERI Inform.*, vol. 9, no. 2, 2024, [Online]. Available: https://jurnal.polbeng.ac.id/index.php/ISI/article/download/204/63/783

[7] N. Sitorus; and E. Al, "Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2," *J. Quancom*, vol. 2, no. 2, 2024, [Online]. Available: https://journal.iteba.ac.id/index.php/jurnal_quancom/article/download/432/235/2787

[8] I. Rahim; and et al, "Komparasi Fungsi Hash Md5 Dan Sha256 Dalam Keamanan Gambar Dan Teks," *J. IKRAITH-INFORMATIKA*, vol. 7, no. 2, 2023, [Online]. Available: https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive%0A