Jurnal Teknologika

Journal homepage:
https://jurnal.wastukancana.ac.id/index.php/teknologika/index
ISSN: 2715-4645 E-ISSN: 1693-2978

Analisis Komparatif Algoritma SHA-256 dan BLAKE2: Studi Kasus Efisiensi Waktu Proses, Penggunaan CPU, dan RAM pada Sistem Integritas Data di Windows dan Linux

Comparative Analysis of SHA-256 and BLAKE2 Algorithms: A Case Study of Processing Time Efficiency, CPU Usage, and RAM in Data Integrity Sistems on Windows and Linux

Yusuf Muhyidin[1*], Imam Ma'ruf Nugroho Author [2,], Muhammad Agus Sunandar [3]

[1,2,3] Informatics Engineering, STT wastukancana, Purwakarta, Indonesia

Abstark: Keamanan data merupakan aspek fundamental dalam sistem informasi modern, terutama dalam menjamin integritas data yang dikirimkan maupun disimpan. Fungsi kriptografi hash menjadi teknologi kunci dalam memenuhi kebutuhan tersebut, dengan SHA-256 sebagai standar industri yang telah digunakan secara luas, dan BLAKE2 sebagai alternatif yang dirancang untuk menawarkan kecepatan lebih tinggi pada arsitektur prosesor modern. Namun demikian, mayoritas penelitian terdahulu membandingkan kedua algoritma hanya pada satu platform dan terbatas pada parameter waktu proses, tanpa mempertimbangkan pengaruh sistem operasi serta konsumsi sumber daya secara menyeluruh. Penelitian ini bertujuan untuk melakukan analisis komparatif yang komprehensif terhadap kinerja SHA-256 dan BLAKE2 pada dua lingkungan sistem operasi, yaitu Windows 11 dan Linux dengan mengukur empat parameter utama yaitu waktu eksekusi (ms), *throughput* (MB/s), penggunaan CPU (%), dan konsumsi RAM (MB). Pengujian dilakukan menggunakan bahasa pemrograman Python 3.13 dengan library hashlib, timeit, psutil, dan tracemalloc, pada perangkat keras AMD Ryzen 5 5500 dengan RAM 16 GB DDR4. Data uji dihasilkan secara acak melalui os.urandom() dalam empat variasi ukuran, yaitu 1 MB, 10 MB, 50 MB, dan 100 MB, masing-masing diulang sebanyak 100 iterasi untuk memperoleh nilai rata-rata yang stabil. Hasil pengujian menunjukkan bahwa tidak ada algoritma yang secara universal lebih unggul di semua kondisi. Pada lingkungan Linux, BLAKE2 lebih cepat untuk ukuran data 1–50 MB, sedangkan SHA-256 lebih unggul pada data 100 MB. Sebaliknya, pada Windows, SHA-256 lebih cepat untuk data 1–10 MB, sementara BLAKE2 jauh lebih unggul pada data 100 MB dengan waktu 5,10 ms dibanding SHA-256 yang membutuhkan 18,82 ms. Temuan paling signifikan adalah Windows mengonsumsi CPU rata-rata 2,84–3,12%, yang tiga kali lebih rendah dibandingkan Linux sebesar 9,68–9,84%, mengindikasikan perbedaan mendasar dalam manajemen sumber daya komputasi kriptografi antara kedua sistem operasi.
Keywords: Algoritma hashing; SHA-256; BLAKE2; Linux; Windows

Abstract: *Data security is a fundamental aspect of modern information sistems, particularly in ensuring the integrity of transmitted and stored data. Cryptographic hash functions serve as a key technology to meet this need, with SHA-256 as the widely adopted industry standard and BLAKE2 as an alternative designed to offer higher speed on modern processor architectures. However, most previous studies compared both algorithms on a single platform and were limited to processing time parameters, without considering the influence of the operating sistem or overall resource consumption. This study aims to conduct a comprehensive comparative analysis of SHA-256 and BLAKE2 performance on two operating sistem environments, namely Windows 11 and Linux, by measuring four main parameters: execution time (ms), throughput (MB/s), CPU usage (%), and RAM consumption (MB). Testing was conducted using Python 3.13 with the hashlib, timeit, psutil, and tracemalloc libraries, on AMD Ryzen 5 5500 hardware with 16 GB DDR4 RAM. Test data was generated randomly via os.urandom() in four size variations — 1 MB, 10 MB, 50 MB, and 100 MB — each repeated 100 iterations to obtain stable average values. The results show that no algorithm is universally superior under all conditions. On Linux, BLAKE2 is faster for data sizes of 1–50 MB, while SHA-256 outperforms on 100 MB data. Conversely, on Windows, SHA-256 is faster for 1–10 MB data, while BLAKE2 significantly outperforms on 100 MB data with an execution time of 5.10 ms compared to SHA-256's 18.82 ms. The most significant finding is that Windows consumes an average CPU of 2.84–3.12%, which is three times lower than Linux at 9.68–9.84%, indicating a fundamental difference in cryptographic resource management between the two operating sistems.*
Keywords: *Hashing algorithm; SHA-256; BLAKE2; Linux; Windows*

[*] *Corresponding author : yusufmuhyidin@wastukancana.ac.id*
https://doi.org/10.51132/teknologika.v16i1
Received : 19-04-2026 Accepted : 20-04-2026 Available online : 31-05-2026

1. Pendahuluan

Perkembangan teknologi digital telah meningkatkan jumlah serta kompleksitas data yang diproses setiap hari. Aliran informasi kini mengalir deras melalui jaringan internet sebagai infrastruktur utama operasional. Kondisi ini menuntut adanya mekanisme perlindungan data yang mampu menjamin keutuhan informasi. Integritas data menjadi salah satu aspek penting dalam keamanan informasi karena memastikan bahwa data tidak mengalami perubahan tanpa izin. Dalam konteks ini, keamanan data menjadi aspek kritis yang tidak dapat diabaikan oleh organisasi, institusi, maupun individu. Menjamin integritas data menjadi aspek yang tidak dapat diabaikan dalam membangun sistem keamanan informasi yang andal informasi yang dikirimkan ataupun disimpan akan selalu dalam keadaan utuh tidak dirubah ditambah ataupun dikurangi.[1]

Fungsi hash kriptografi digunakan untuk menghasilkan representasi unik dari suatu data. Algoritma yang baik harus mampu menghasilkan output yang konsisten, sulit diprediksi, serta tahan terhadap benturan (*collision*). SHA-256 merupakan salah satu algoritma yang banyak digunakan dalam berbagai sistem keamanan, termasuk komunikasi aman dan teknologi *blockchain*.[2] Beberapa penelitian menunjukkan bahwa meskipun SHA512 memiliki tingkat kerumitan yang lebih tinggi dan ukuran hash yang lebih panjang, dalam konteks tertentu seperti efisiensi dan kecepatan pemrosesan pada perangkat tertentu, SHA-256 dianggap lebih optimal.[3] Karakteristik SHA-256 yang cukup berat secara komputasi mendorong munculnya inovasi algoritma baru seperti BLAKE2. BLAKE2 dirancang untuk menawarkan keamanan yang setara dengan SHA-256 namun dengan kecepatan proses yang jauh lebih tinggi, terutama pada arsitektur prosesor modern 64-bit.

Dalam konteks industri keuangan modern, seperti pada sistem *high-frequency trading* atau verifikasi aset kripto, keterlambatan (*latency*) dalam hitungan milidetik akibat proses hashing yang berat dapat menyebabkan kerugian finansial yang signifikan. Di sisi lain, pada implementasi perangkat *Internet of Things* (IoT) dengan daya komputasi rendah, penggunaan SHA-256 sering kali menguras baterai dan memori secara berlebihan. Fenomena ini menunjukkan bahwa keamanan tanpa efisiensi kecepatan tidak lagi cukup; dibutuhkan solusi seperti BLAKE2 yang mampu memberikan perlindungan setara namun tetap ringan, guna mencegah *bottleneck* pada performa sistem keamanan di masa depan.[4] Pada penelitian sebelumnya menunjukkan bahwa BLAKE2 memiliki keunggulan dalam hal kecepatan, sementara SHA-256 tetap unggul dalam hal standardisasi dan adopsi luas. Namun demikian, parameter yang dianalisis hanya berfokus pada waktu proses tanpa mempertimbangkan penggunaan sumber daya sistem secara menyeluruh, seperti penggunaan CPU dan memori (RAM). Selain itu, faktor lingkungan eksekusi seperti sistem operasi juga jarang dijadikan variabel penelitian, padahal perbedaan sistem operasi dapat mempengaruhi performa algoritma secara signifikan

Berdasarkan permasalahan tersebut, maka perlu dilakukan penelitian yang bersifat komparatif untuk menguji secara mendalam efektivitas dan efisiensi dari algoritma SHA-256 dan BLAKE2. dengan mempertimbangkan beberapa parameter, yaitu waktu proses, penggunaan CPU, dan penggunaan RAM dengan berbagai ukuran data, serta membandingkan performanya pada dua sistem operasi, yaitu Windows dan Linux.

1.1 Konsep Keamanan Informasi

Tiga Pilar Keamanan Informasi (CIA Triad) [5]:

1. *Confidentiality* (Kerahasiaan)

Kerahasiaan adalah upaya melindungi informasi agar hanya dapat diakses oleh pihak yang berwenang. Tujuan utama kerahasiaan adalah memastikan bahwa data sensitif tidak jatuh ke tangan yang salah atau tidak sah [6].

2. *Integrity* (Integritas)
   Integritas adalah kemampuan untuk memastikan bahwa informasi tetap akurat, konsisten, dan tidak diubah tanpa izin. Tujuan integritas adalah melindungi informasi dari modifikasi yang disengaja maupun tidak disengaja.
3. *Availability* (Ketersediaan)
   Ketersediaan adalah memastikan bahwa informasi dapat diakses oleh pihak yang berwenang kapan pun informasi tersebut dibutuhkan. Tujuan utama ketersediaan adalah mencegah gangguan terhadap akses informasi yang diperlukan, terutama dalam situasi kritis.

1.2 SHA-256

SHA-256 (*Secure Hash Algorithm 256-bit*) adalah algoritma hashing dalam keluarga SHA-2 yang menawarkan tingkat keamanan lebih tinggi dibandingkan MD5. Algoritma ini menghasilkan *output* dengan panjang tetap 256-bit, menjadikannya lebih tahan terhadap serangan tabrakan. Namun, SHA-256 membutuhkan waktu komputasi lebih lama dibandingkan MD5.[7]

1.3 BLAKE2

Algoritma BLAKE2 dikembangkan sebagai respons terhadap kebutuhan sistem komputasi modern yang memerlukan fungsi hash yang cepat, efisien, dan tetap aman. Dirancang untuk arsitektur 64-bit, BLAKE2 mampu menghasilkan nilai hash sepanjang 512-bit dan telah dioptimalkan untuk performa tinggi di berbagai platform, termasuk perangkat *embedded* dan sistem terdistribusi. Algoritma ini merupakan turunan dari BLAKE, yang sebelumnya menjadi finalis dalam kompetisi SHA-3 yang diadakan oleh NIST.[8]

1.4 Fungsi Hash

Fungsi hash dalam kriptografi digunakan untuk mengonversi data dengan ukuran bervariasi menjadi representasi tetap yang unik dan tidak dapat dibalik. Algoritma hash seperti SHA-1, SHA-256, dan SHA-512 banyak digunakan untuk menjamin integritas data, karena perubahan sekecil apa pun pada data akan menghasilkan nilai hash yang berbeda secara signifikan. [9]

1.5 Sistem Operasi

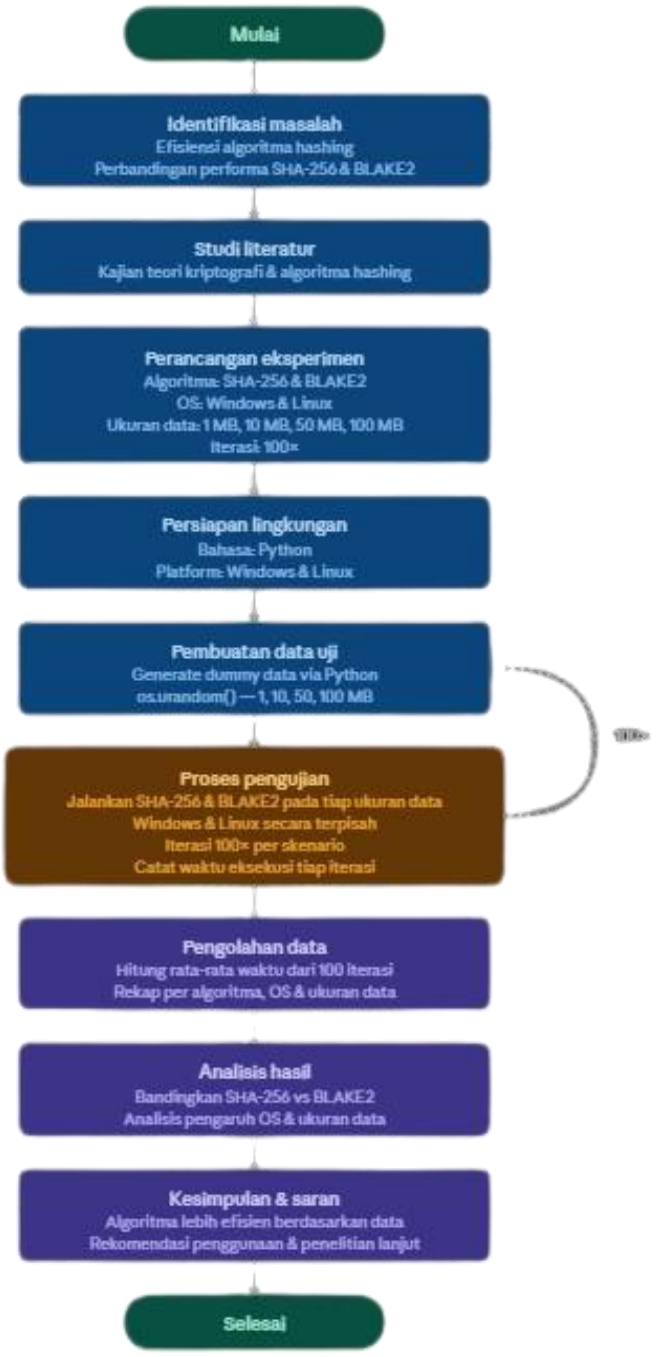
Sistem Operasi Adalah seperangkat program yang mengelola sumber daya perangkat keras komputer, dan menyediakan layanan umum untuk aplikasi perangkat lunak.

1. *Linux*
   Linux adalah sistem operasi berbasis GNU/Linux yang bersifat *Open Source* dan memiliki banyak varian seperti Debian, Slackware, Open Suse, Archlinux, Redhat dan sebagainya yang diciptakan oleh Linus Trovald.[12]
2. *Windows*
   Microsoft Windows merupakan salah satu sistem operasi paling populer di dunia yang banyak digunakan pada perangkat laptop, PC, hingga server. Keberadaannya menjadi fondasi utama dalam menjalankan berbagai aplikasi, mulai dari pekerjaan kantor, belajar, hiburan, hingga kebutuhan kreatif.

2. Metodologi

2.1 Alur Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sistematis sebagai berikut:



Gambar 1. Alur Penelitian

1. Studi Literatur: Mengumpulkan referensi terkait algoritma SHA-256 dan BLAKE2.
2. Persiapan Lingkungan: Menyiapkan perangkat keras dan instalasi Python.
3. Perancangan Skenario: Menentukan ukuran data uji dan jumlah iterasi.
4. Ekperimen/Pengujian: Menjalankan kode program pengujian kecepatan.
5. Pengumpulan Data: Mencatat hasil waktu eksekusi dan throughput.
6. Analisis & Penarikan Kesimpulan: Membandingkan hasil dan menyusun laporan.

2.2 Lingkungan Pengujian

Untuk menjaga konsistensi hasil, pengujian dilakukan pada lingkungan yang terkontrol:

Tabel 1. Spesifikasi Perangkat Yang Digunakan

Komponen	Spesifikasi
Processor	AMD Ryzen 5 5500 with Radeon Graphics
Memory RAM	16 GB DDR4 4800 MHz
Sistem Operasi	Windows 11 (64Bit)
Bahasa Pemrograman	Python 3.13.7
Library Python	hashlib, timeit, os
Dataset (Dummy)	1 MB, 10 MB, 50MB, 100 MB
Jumlah Iterasi	100 kali per pengujian

2.3 Skenario Pengujian

Data uji dihasilkan secara acak melalui fungsi os.urandom() untuk memastikan variasi bit data.

1. Ukuran Dummy Data: 1 MB, 10 MB, dan 50 MB 100MB.
2. Iterasi: Setiap pengujian diulang sebanyak 100 kali.
3. Metode Pengukuran: Menggunakan modul timeit untuk mendapatkan akurasi waktu hingga tingkat mikrodetik.

   Dalam Python, cara tercepat dan paling akurat untuk membuat data sampah (dummy data) guna keperluan riset kriptografi adalah dengan menggunakan fungsi os.urandom(). Fungsi ini menghasilkan byte acak yang tidak dapat dikompresi, sehingga simulasi enkripsi/hashing menjadi lebih realistis.

2.4 Parameter Penilaian

1. Execution Time (ms): Total waktu yang dibutuhkan algoritma untuk menghasilkan nilai hash dari input data tertentu.
2. Throughput (MB/s): Kecepatan pemrosesan data yang dihitung dengan
3. Penggunaan CPU (%)
4. Penggunaan RAM (MB)

2.5 Prosedur Pengujian (Psudo-code)

1. Inisialisasi dataset dengan ukuran tertentu.
2. Jalankan fungsi hash SHA-256, catat waktu mulai dan waktu selesai.
3. Jalankan fungsi hash BLAKE2, catat waktu mulai dan waktu selesai.
4. Hitung rata-rata waktu dari 100 kali pengulangan.
5. Simpan hasil ke dalam tabel perbandingan.

2.6 Skema pengujian

Untuk mendapatkan data yang akurat, kita akan menggunakan skema berikut:

1. Variabel Bebas: Algoritma (SHA-256 vs BLAKE2b).
2. Variabel Terikat: Waktu eksekusi (milidetik), *Throughput* (MB/detik), Penggunaan CPU(%) dan RAM (MB).
3. Variabel Kontrol: Ukuran data yang sama, perangkat keras yang sama, dan jumlah pengulangan (*loop*) yang sama.
4. Skenario: Kita akan menguji tiga ukuran data: Small (1 MB), Medium (10 MB), Large (50 MB dan Extra Large (100 MB) untuk melihat apakah perbedaan kecepatan semakin lebar saat data membesar.

3. Hasil dan Pembahasan

Dalam pengujian algoritma SHA-256 dan BLAKE2 penulis menggunakan libarary os.urandom dimana data yang akan diuji dibuat acak, ini sangat disarankan daripada teks biasa (seperti "aaaaa...").

1. Alasan Keamanan: Data acak menguji seluruh logika matematis algoritma secara maksimal.
2. Alasan Realistis: Data acak menyerupai file asli seperti gambar, video, atau dokumen terenkripsi.

Pengujian dilakukan untuk membandingkan performa algoritma SHA-256 dan BLAKE2 berdasarkan parameter waktu eksekusi, throughput, penggunaan CPU, dan penggunaan RAM. Pengujian dilakukan pada dua sistem operasi, yaitu Windows dan Linux, dengan ukuran data 1 MB, 10 MB, 50 MB, dan 100 MB. Setiap skenario pengujian dilakukan sebanyak 100 iterasi untuk memperoleh nilai rata-rata yang lebih stabil dan mengurangi pengaruh fluktuasi sistem.

Bahasa pemrograman yang digunakan dalam penelitian ini adalah python 3.13 berikut library yang digunakan.

```
import hashlib
import time
import os
import psutil
import csv
import platform
import tracemalloc
```

Gambar 2. Library Python

3.1 Hasil Pengujian Pada Sistem Operasi Linux

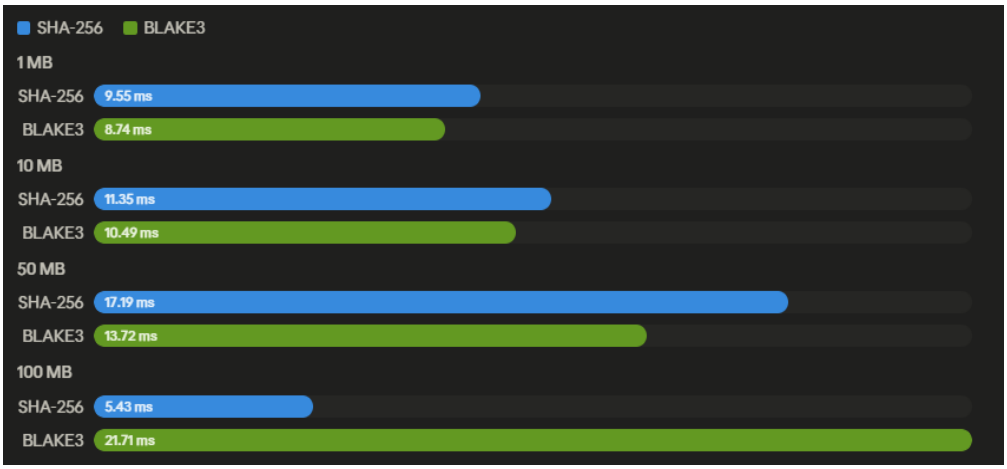
Pada tabel dibawah ini merupakan nilai rata-rata dari hasil pengujian iterasi sebanyak 100x.

Tabel 3. Hasil Pengujian Pada Lingkungan Linux

Ukuran Data	Algoritma	Waktu (ms)	Throughput (MB/s)	CPU (%)	RAM (MB)
1 MB	SHA-256	9.55	7.99	9.23	0.0324
	BLAKE2	8.74	7.81	9.30	0.0323
10 MB	SHA-256	11.34	9.28	9.63	0.0324
	BLAKE2	10.48	9.07	10.18	0.0323
50 MB	SHA-256	17.18	11.90	10.19	0.0324
	BLAKE2	13.72	13.38	9.85	0.0323
100 MB	SHA-256	5.42	13.44	9.66	0.0324
	BLAKE2	21.71	16.47	10.01	0.0323

Berdasarkan hasil pengujian sebanyak 100 iterasi dengan variasi data 1MB, 10MB, 50MB dan 100MB, diperoleh nilai rata-rata untuk setiap parameter. Misal pada tabel diatas dengan jumlah data sebesar 1MB pada algoritma SHA-256 waktu yang dibutuhkan untuk proses *hashing* adalah 9.55ms dan sistem mampu memproses 7.99MB perdetik dan algoritma menggunakan kemampuan CPU sebesar 9.23% dengan penggunaan RAM sebesar 0.0323MB saat proses berlangsung.

3.2 Grafik Hasil Perbandingan



Gambar 5. Perbandingan Waktu Eksekusi (MS) Per Ukuran File

Gambar 6. Throughput Per Ukuran File (MBPS)

Dari hasil pengujian tersebut didapatkan beberapa temuan berikut diantaranya:

1. BLAKE2 lebih cepat di file kecil–menengah (1–50 MB) dengan selisih 0.81–3.46 ms. Namun SHA-256 jauh lebih unggul di file 100 MB (5.43 ms vs 21.71 ms).
2. Throughput BLAKE2 lebih tinggi di file besar (50–100 MB) — mencapai 16.47 MBps vs 13.45 MBps SHA-256 pada 100 MB, menunjukkan efisiensi data yang lebih baik saat ukuran besar.
3. Penggunaan CPU dan RAM hampir identik antara kedua algoritma. CPU rata-rata 9.68% (SHA) vs 9.84% (BLAKE), RAM keduanya ~0.0323–0.0324 MB.

3.3 Hasil Pengujian Pada Sistem Operasi Windows

Pada tabel dibawah ini merupakan nilai rata-rata dari hasil pengujian iterasi sebanyak 100x

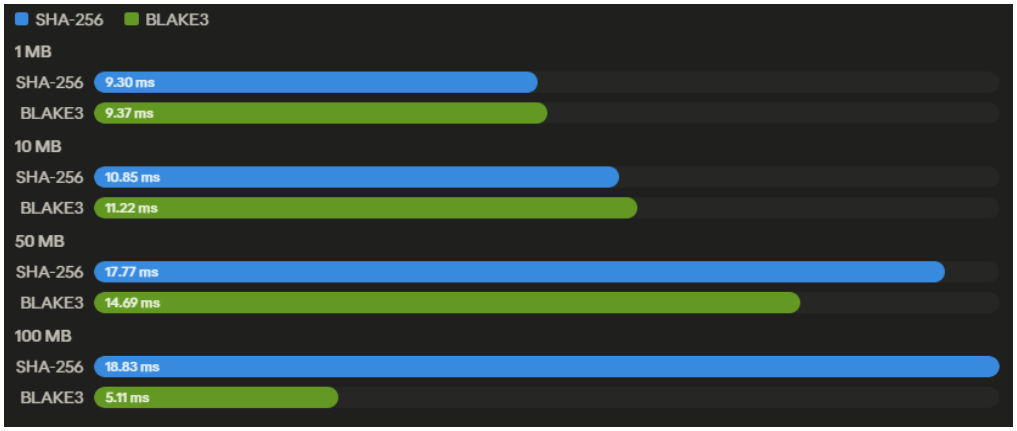
Tabel 2. Hasil Pengujian Pada Lingkungan Windows

Ukuran Data	Algoritma	Waktu (ms)	Throughput (MB/s)	CPU (%)	RAM (MB)
1 MB	SHA-256	9.29	8.27	4.81	0.0166
	BLAKE2	9.36	7.72	5.91	0.0170
10 MB	SHA-256	10.85	6.80	2.34	0.0166
	BLAKE2	11.21	6.24	1.75	0.0170
50 MB	SHA-256	17.77	16.87	1.71	0.0166
	BLAKE2	14.68	11.70	2.03	0.0170
100 MB	SHA-256	18.82	13.48	2.49	0.0166
	BLAKE2	5.10	14.40	2.78	0.0170

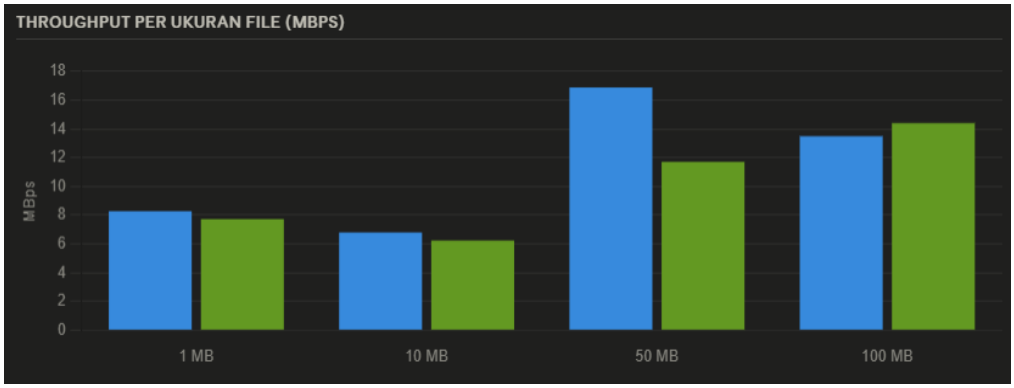
Berdasarkan hasil pengujian sebanyak 100 iterasi dengan variasi data 1MB, 10MB, 50MB dan 100MB, diperoleh nilai rata-rata untuk setiap parameter. Misal pada tabel diatas dengan jumlah data sebesar 1MB waktu yang dibutuhkan untuk proses *hashing* adalah 9.29ms sistem mampu memproses 8.27MB perdetik dan algoritma menggunakan kemampuan CPU sebesar 4.81% dan penggunaan RAM sebesar 0.0116MB saat proses berlangsung.

Nilai waktu menunjukkan kecepatan proses, *throughput* menunjukkan efisiensi pemrosesan data, sedangkan CPU dan RAM menunjukkan penggunaan sumber daya sistem. Penggunaan memori diukur menggunakan metode peak memory dengan bantuan *library tracemalloc* untuk memperoleh nilai penggunaan memori maksimum selama proses *hashing*.

3.4 Grafik Hasil Perbandingan



Gambar 3. Perbandingan Waktu Eksekusi (MS) Per Ukuran File



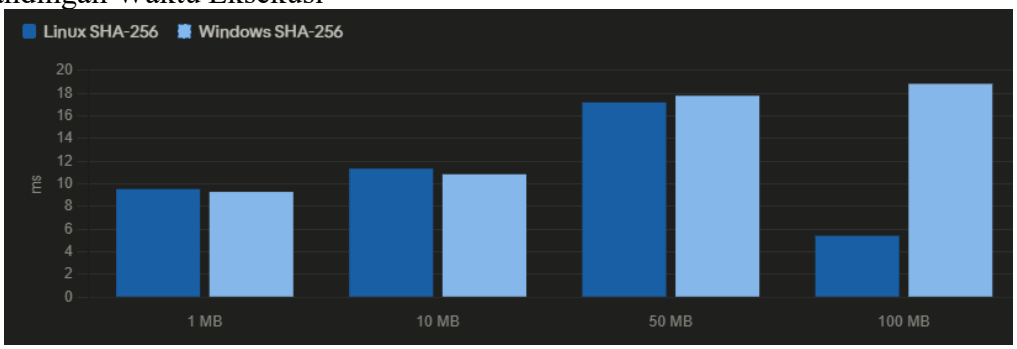
Gambar 4. *Throughput* Per ukuran File (MBPS)

Dari hasil pengujian tersebut didapatkan beberapa temuan berikut diantaranya:

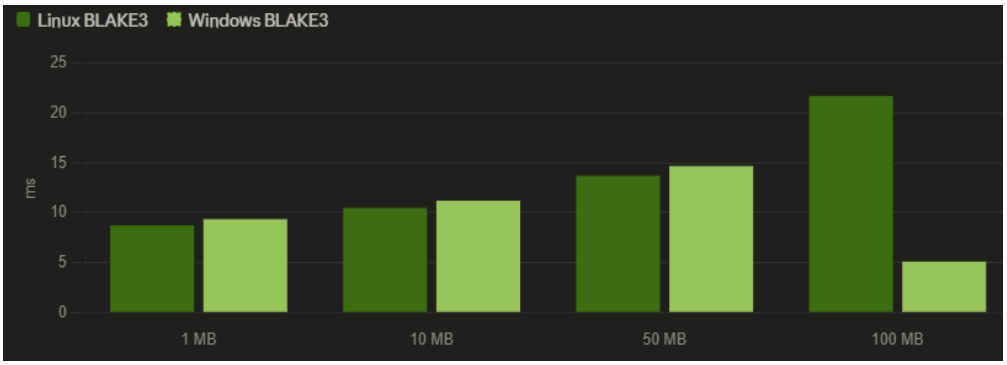
1. SHA-256 lebih cepat di file kecil–menengah (1–10 MB) di Windows, berbeda dengan Linux di mana BLAKE2 unggul pada rentang yang sama.
2. BLAKE2 jauh lebih unggul di file 100 MB — hanya 5.11 ms vs SHA-256 18.83 ms (3.7× lebih cepat). Ini kebalikan dari pola di Linux.
3. CPU Windows jauh lebih rendah dibanding Linux — rata-rata 2.84% (SHA) dan 3.12% (BLAKE), sekitar 3× lebih hemat CPU dari Linux (~9.7%).
4. RAM Windows sedikit lebih tinggi (~0.017 MB vs ~0.032 MB di Linux), namun SHA lebih hemat RAM dibanding BLAKE2 di Windows.

3.5 Hasil Perbandingan pada 2 sistem operasi

1. Perbandingan Waktu Eksekusi

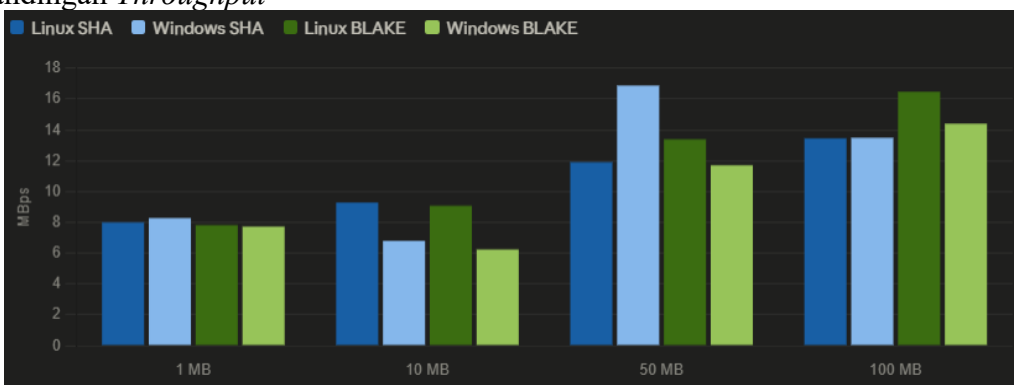


Gambar 5. Perbandingan Waktu Eksekusi SHA-256 Pada Linux & Windows



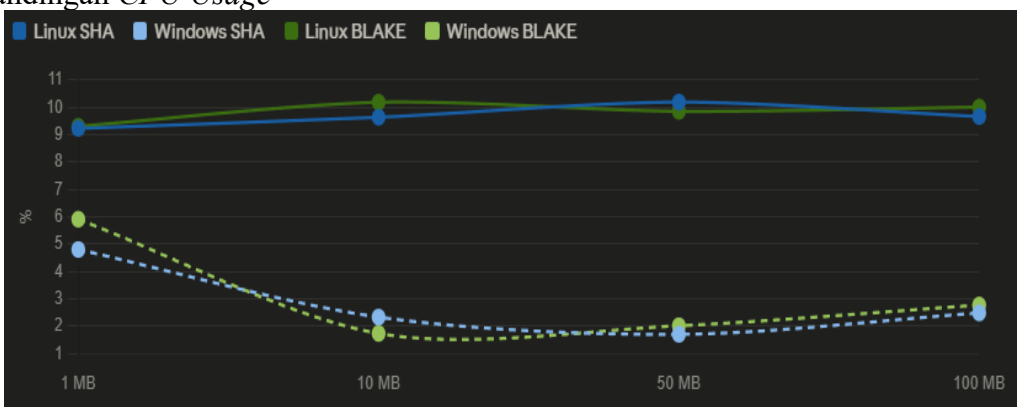
Gambar 6. Perbandingan Waktu Eksekusi BLAKE2 Pada Linux & Windows

2. Perbandingan *Throughput*



Gambar 7. Throughput Semua Kombinasi (MBPS)

3. Perbandingan *CPU Usage*



Gambar 8. CPU usage Rata-rata per ukuran File (%)

4. Kesimpulan

4.1 Tidak ada algoritma yang unggul disemua kondisi

Pernyataan bahwa BLAKE2 lebih unggul dari SHA-256 ataupun sebaliknya tidak didukung data saat penelitian ini dilakukan. keunggulan bergantung pada kondisi: Ukuran file, sistem operasi dan parameter yang digunakan (Waktu, *Throughput*, CPU dan RAM). Jika dilingkungan linux dengan File ukuran 1MB-10MB lebih baik gunakan BLAKE2 karena lebih cepat 8-85%. jika ukuran file besar 100MB keatas gunakan SHA-256 karena lebih cepat. sebaliknya jika pada lingkungan windows file ukuran 1MB-10MB lebih cocok menggunakan SHA-256 dan ukuran 50MB-100MB gunakan BLAKE2.

4.2 Windows Hemat CPU Linux Untuk beban besar

Jika efesiensi energi menjadi pertimbangan contohnya Server dengan beban tinggi windows menunjukan penggunaan CPU 3-8x lebih rendah dibanding linux.

4.3 Temuan penelitian

1. BLAKE2 lebih dominan pada lingkungan sistem operasi Linux untuk ukuran file kecil SHA-256 dominan di Windows untuk ukuran file kecil
2. Metrik CPU lebih efisien pada sistem operasi Windows.

Daftar Pustaka

[1] Octavian, A. (2020). Keamanan Maritim dan Tantangan Pertahanan Siber di Indonesia. Universitas Pertahanan RI

[2] Aria Bagas dkk (2025). Analisis Penggunaan *Hash Function* dan *Digital Signature* pada Sistem Keamanan Informasi

[3] A Yoshida dkk dalam Toras Batubara (2025). Perbandingan Algoritma Kriptografi Hash Sha 256 dan Sha 512 Untuk Keamanan Sandi

[4] Ahmad & Fikri (2025). Analisis Performa Algoritma BLAKE2 dan SHA-256 pada Implementasi *Blockchain*

[5] Indarta, Y., Ranuhardja, F., & Ashari, I. F. (2022). Keamanan Siber Tantangan di Era Revolusi Industri 4.0

[6] Ashari, I. F. et al. (2024) Dasar-dasar Keamanan Informasi. Yayasan Kita Menulis

[7] Novin dkk (2024). Analisis Kinerja Algoritma MD5, SHA-256, dan Base62 dalam Sistem Pemendekan URL

[8] A. M. Fajrin, (2023) Perbandingan performa kecepatan dari algoritma hash function untuk proses enkripsi password, ‖ Kesatria: Jurnal Penerapan Sistem Informasi (Komputer dan Manajemen), vol. 4, no. 4, pp. 1069–1075

[9] Ipdal, M. (2021). Analisa metode SHA-512 untuk tanda tangan digital pada file video. *Journal of Informatics Management and Information Technology*, 1(1), 23–29.

[10] Yahfizam (2023). Studi Literatur Perbandingan Bahasa Pemrograman C++ Dan Bahasa Pemrograman Python Pada Algoritma Pemrograman. Jurnal Teknik Informatika dan Teknologi Informasi (JUTITI)Vol. 3 No. 3

[11] Tri Listyorini (2013). Perancangan Mobile Learning Mata Kuliah Sistem Operasi Berbasis Android. Jurnal SIMETRIS, Vol 3 No 1 April 2013

[12] Edy Budi Harjono (2016) Analisa Dan Implementasi Dalam Membangun Sistem Operasi Linux Menggunakan Metode LSF Dan REMASTER. Jurnal & Penelitian Teknik Informatika. Volume 1 Nomor 1, Oktober 2016.