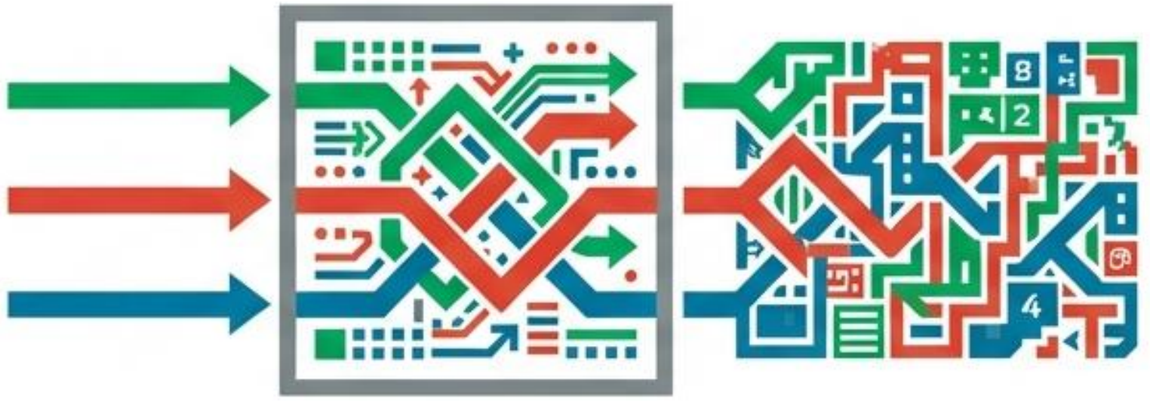
Analisis Komparatif Performa dan Keamanan Algoritma Hash SHA-256, SHA-3, dan BLAKE2 pada Lingkungan Python

Evaluasi Throughput, Efisiensi CPU, dan Strict Avalanche Criterion (SAC) berbasis Arsitektur x86_64

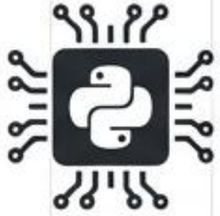
Falito Eriano Nainggolan
Hinggil Parahita
Raffelino Hizkia Marbun
Yosapat Nainggolan



Kesenjangan Antara Teori Asimptotik dan Realita Sistem

O_N Teori & Ekspektasi

- **Standar Baru**: Transisi industri ke SHA-3 (Keamanan) dan BLAKE2 (Kecepatan).
- **Desain Asimptotik**: BLAKE2 dirancang untuk kecepatan software O(n) superior.



Realita di Python

- **Distorsi Interpreter**: Overhead runtime mengaburkan efisiensi algoritmik murni.
- **Instruksi Hardware**: Pengaruh Instruction Set Architecture (ISA) sering diabaikan.

Problem Statement: Apakah teori Big-O berlaku linear di lingkungan interpreted? Diperlukan data empiris.

Rumusan Masalah & Batasan Lingkungan Uji

RQ1: Throughput

Bagaimana komparasi efisiensi SHA-256 vs SHA-3 vs BLAKE2?

RQ2: Hardware Acceleration

Seberapa signifikan dampak instruksi Intel SHA-NI?

RQ3: Difusi (SAC)

Apakah kecepatan mempengaruhi kualitas pengacakan bit?

```
// Batasan (Scope)
Runtime : Python 3.14.2 (hashlib/OpenSSL) | Arsitektur : x86_64 (Intel Core i5-12450H) | Dataset : In-memory generation (1 MB – 1 GB)
```

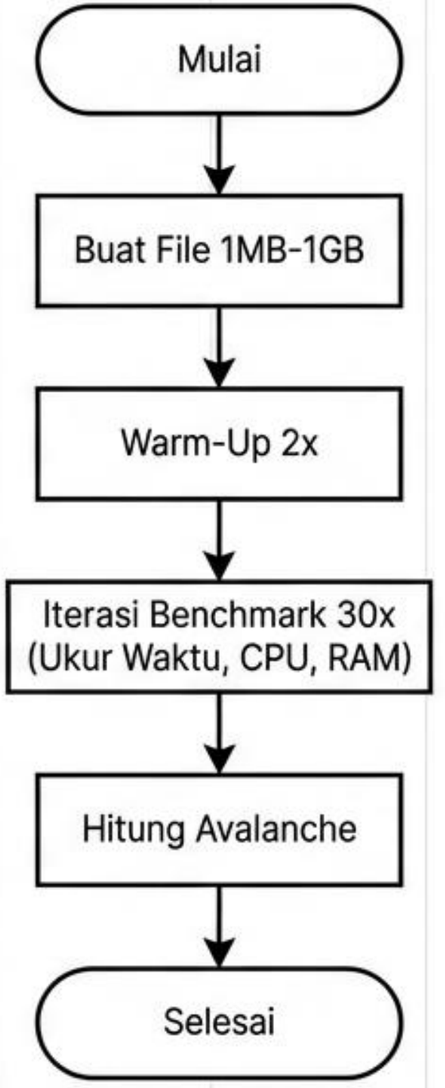
Metodologi: Desain Faktorial 3×4

Desain Eksperimen:

- Faktor 1: Algoritma (3 Level)
- Faktor 2: Ukuran File (4 Level)
- Sampel Total: N = 360
- Iterasi: 30 per sel perlakuan

Kontrol Variabel:

- Warm-up Phase (2x iterasi)
- Time.perf_counter_ns() (Monotonic Clock)



Kerangka Analisis Statistik

Uji Hipotesis Utama

Two-Way ANOVA

Analisis Varians untuk mengukur efek interaksi antara Jenis Algoritma dan Ukuran File.

Validasi & Post-hoc

Bonferroni Adjustment

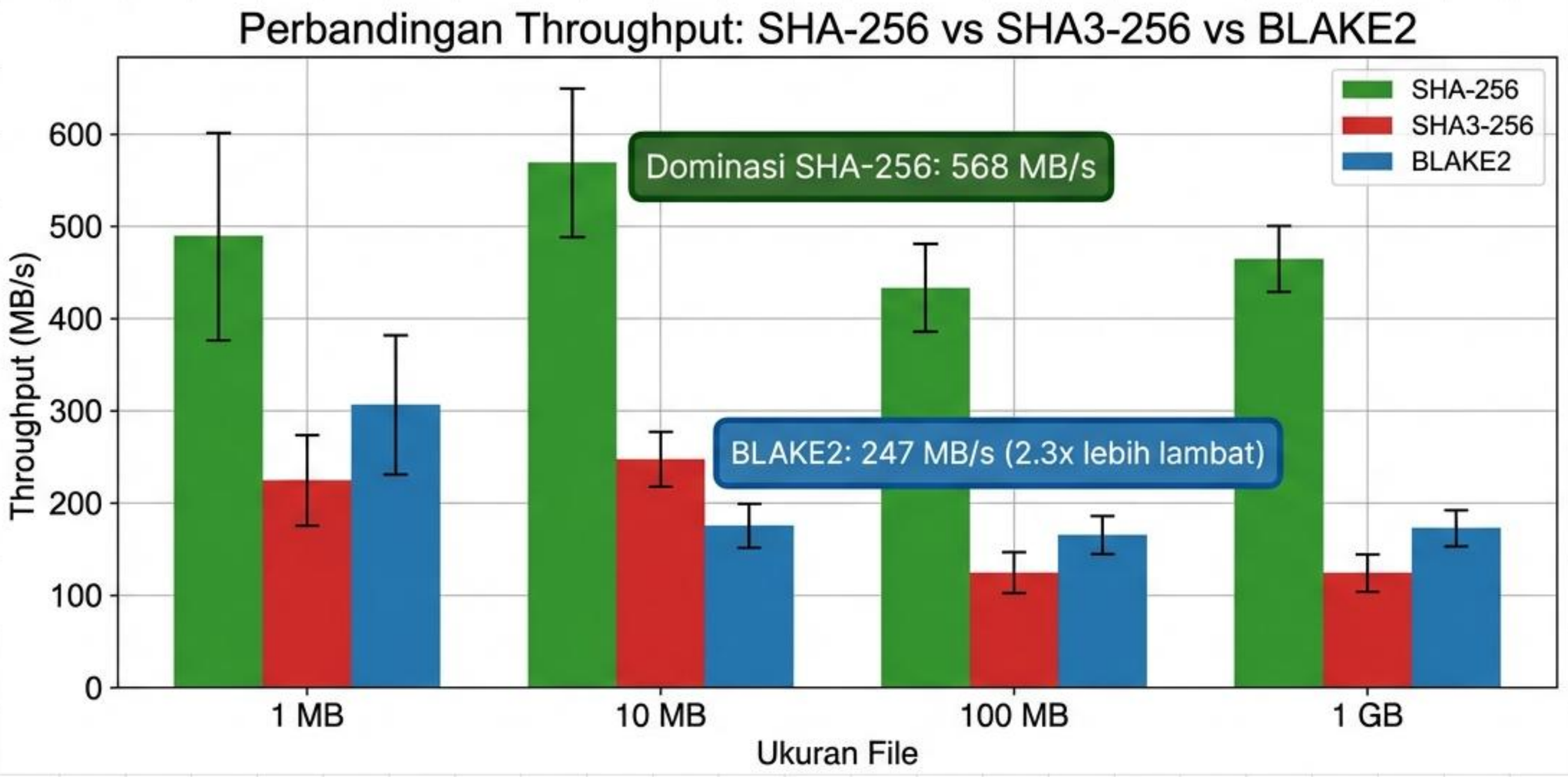
Confidence Interval 95% dengan n=30 per grup (Central Limit Theorem).

Metrik Keamanan

Chi-Square Goodness-of-Fit

Komparasi distribusi bit aktual vs Distribusi Binomial B(256, 0.5).

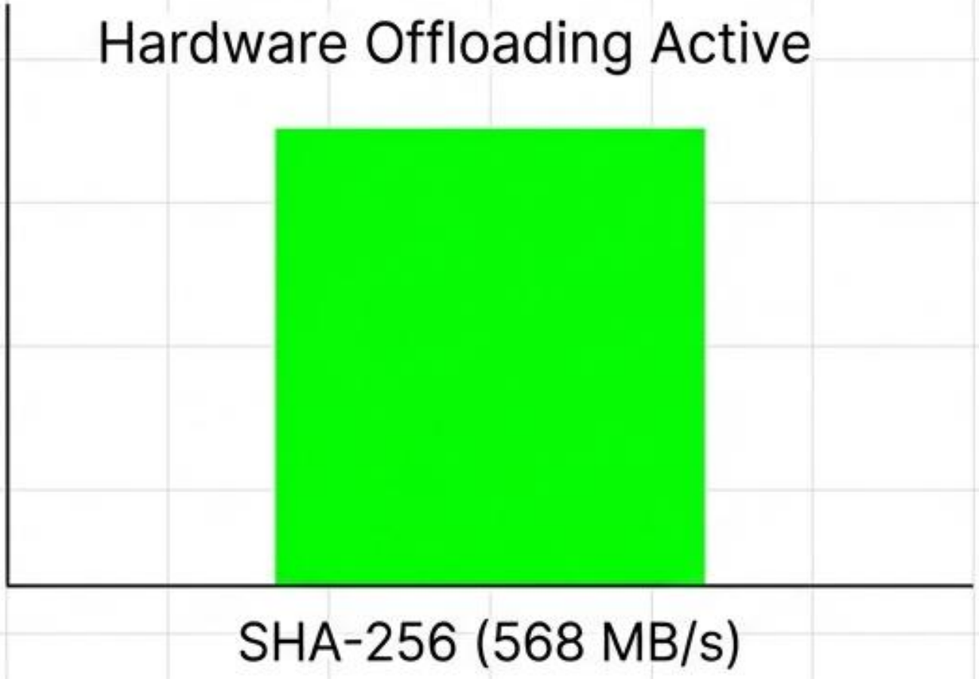
Hasil 1: Evaluasi Throughput (Default Environment)



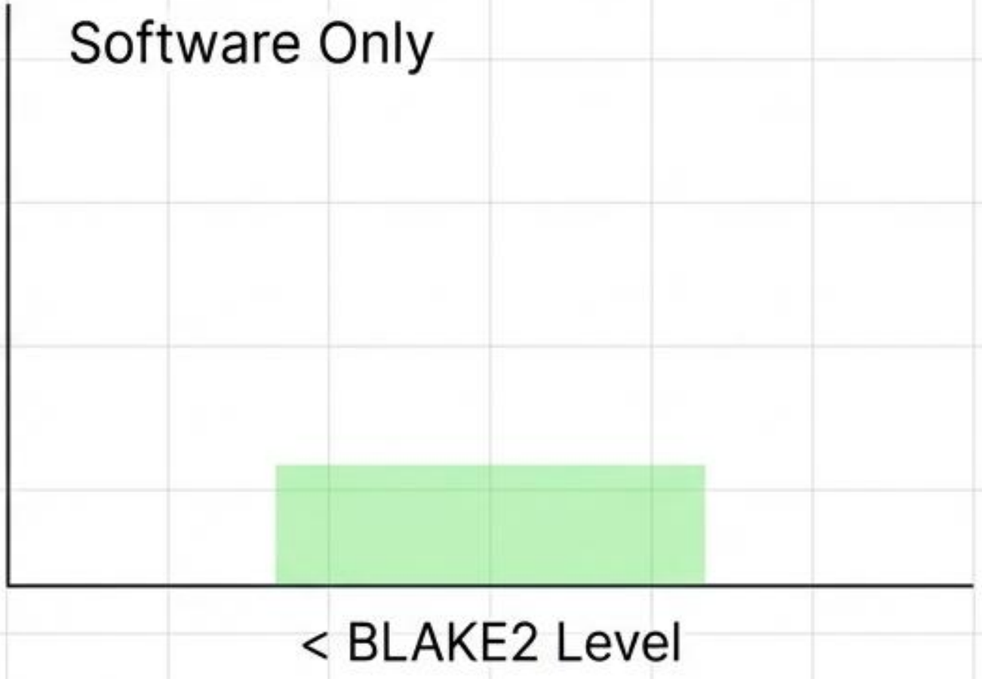
Anomali: Algoritma lama (SHA-2) mengungguli algoritma modern pada lingkungan terakselerasi.

Isolasi Variabel: Dampak Intel SHA Extensions (SHA-NI)

Skenario A: SHA-NI Enabled (Default)



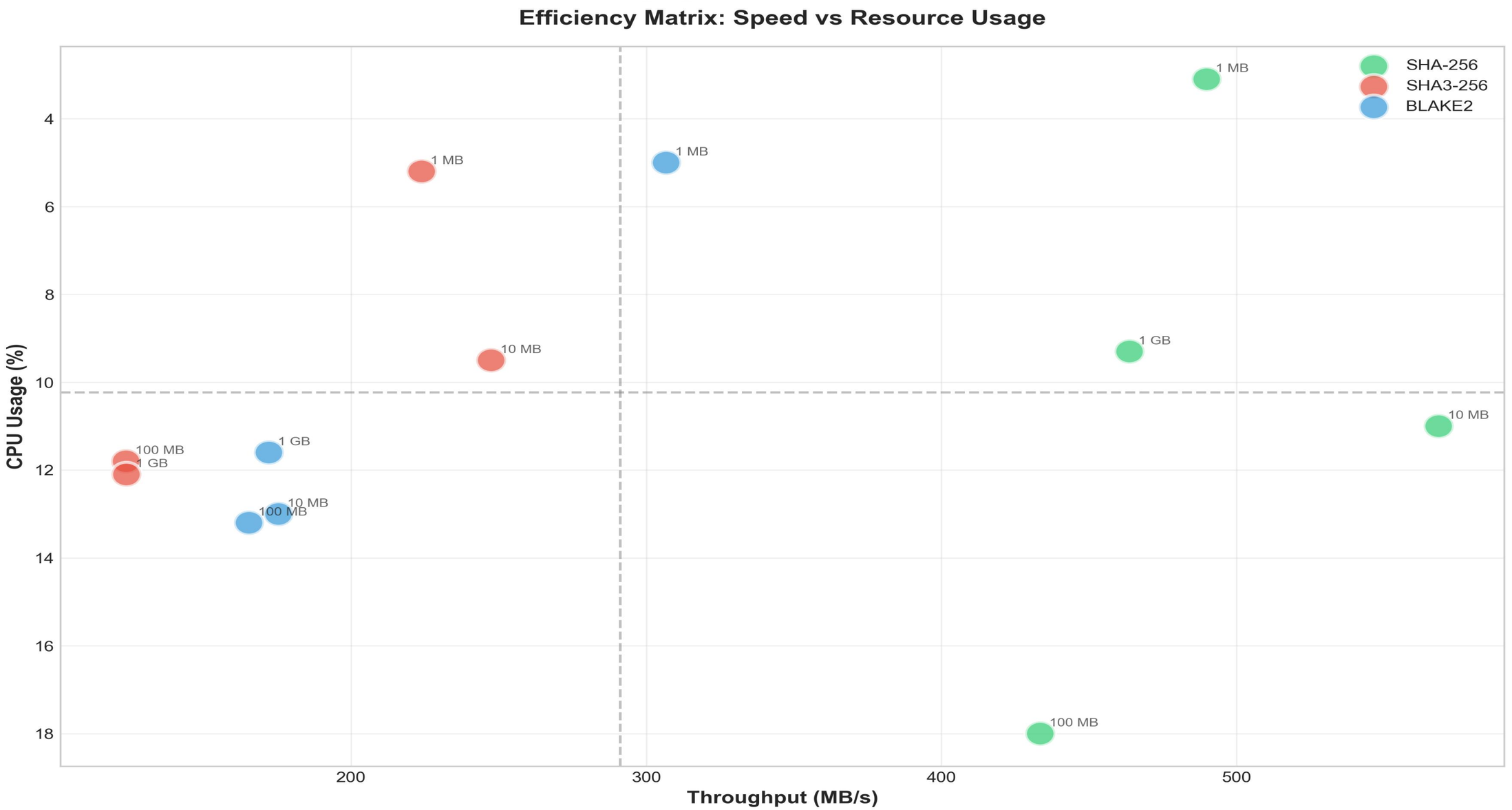
Skenario B: SHA-NI Disabled



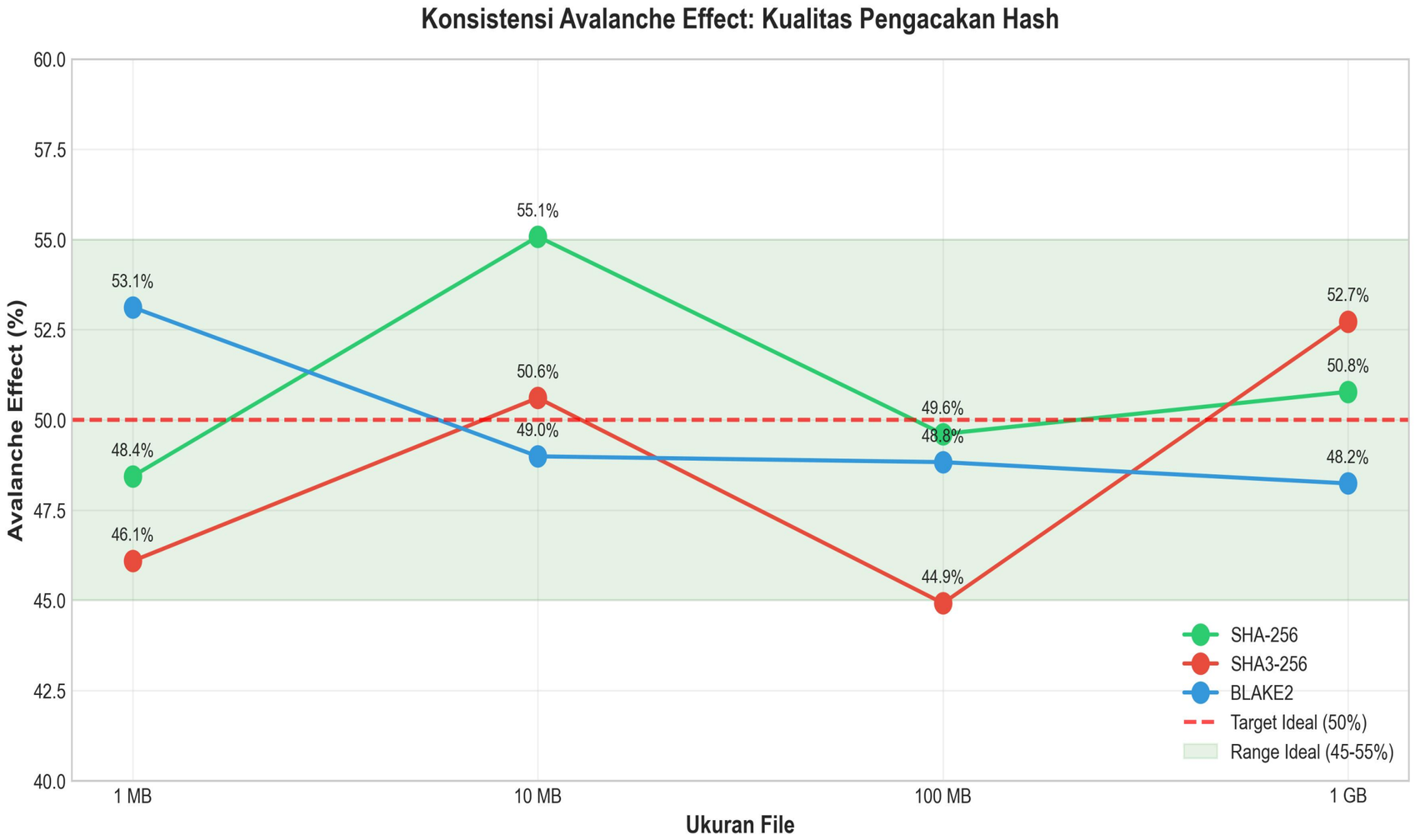

```
OPENSSL_ia32cap='~0x20000000'
```

Kesimpulan Teknis: Keunggulan SHA-256 bukan karena efisiensi kode Python, melainkan dukungan instruksi silikon.

Matriks Efisiensi Penggunaan CPU



Validasi Keamanan: Strict Avalanche Criterion (SAC)



Metodologi:
10.000 bit-flipping iterations.

Hasil:
Hamming Distance ≈ 128 bit (50%).

Statistik:
Chi-Square $p > 0.05$ (H0 Diterima).

Kecepatan tinggi SHA-256 tidak mengorbankan kualitas difusi bit.

Validasi Statistik: Two-Way ANOVA

0.88

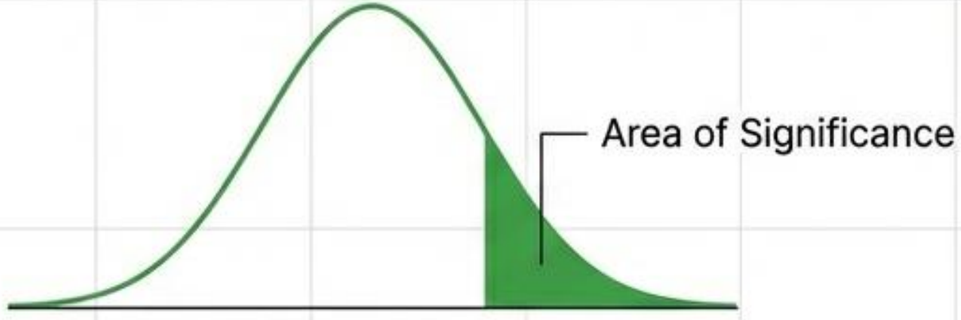
Partial Eta-Squared (ηp^2)

Effect Size Massive: 88% varians performa dijelaskan oleh interaksi Algoritma & Ukuran File.

F(6, 348) = 210.45

F-Score ($p < 0.001$)

Perbedaan sangat signifikan secara statistik.



Diskusi & Implikasi Praktis

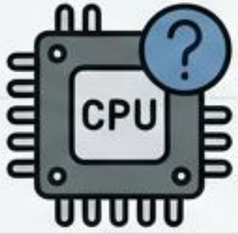
Use SHA-256

Condition
Modern x86_64 Servers

Reason
Throughput maksimal via SHA-NI.

Use BLAKE2

Condition
Legacy Hardware / Low-end



Reason
Jika CPU tidak mendukung ekstensi SHA.

Use SHA-3

Condition
Compliance Only

Reason
Wajib FIPS 202 (High CPU Cost).

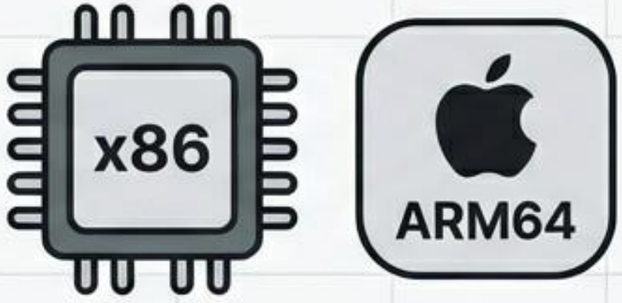
→ **Di Python, algoritma dengan instruksi native (SHA-NI) mengalahkan optimasi software-only.**

Kesimpulan & Future Work

Ringkasan Temuan

- SHA-256 Tercepat (Didorong **SHA-NI**).
- SHA-3 Membebani CPU (Struktur **Sponge**).
- Semua algoritma valid memenuhi **SAC**.

Limitasi & Riset Lanjut



Limitasi: Hasil valid untuk x86_64; tidak dapat digeneralisasi ke ARM64 (Apple Silicon).

Riset Lanjut: Uji pada arsitektur ARM dan pengukuran konsumsi energi (Watt).

Terima Kasih. Sesi Tanya Jawab.