

**Analisis Performa dan Keamanan Algoritma HMAC-SHA-2 dan KMAC-SHA-3
dalam *Time-based One Time Password* pada *Two-Factor Authentication***



SANDY ARDANU BAGASRAGA

NIM. 1910511054

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAKARTA

2026

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas segala rahmat-Nya sehingga Tugas Akhir ini berhasil diselesaikan. Tugas Akhir ini dilakukan dalam bentuk skripsi. Tugas Akhir ini dilaksanakan sejak bulan April 2026 sampai bulan Mei 2026 dengan judul “Analisis Performa dan Keamanan Algoritma HMAC-SHA-2 dan KMAC-SHA-3 dalam *Time-based One Time Password* pada *Two-Factor Authentication*”. Terima kasih penulis ucapkan kepada:

1. Kedua orang tua serta seluruh keluarga yang telah memberikan dukungan doa, dan kasih sayangnya.
2. Prof. Dr. Ir. Supriyanto, ST., M.Sc., IPM selaku Dekan Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta.
3. Dr. Ridwan Raafi'udin, S.Kom., M.Kom selaku Koordinator Program Studi Sarjana Jurusan Informatika Fakultas Ilmu Komputer UPN Veteran Jakarta.
4. Dr. Widya Cholil, S.Kom., M.I.T selaku Dosen Pembimbing Akademik Program Studi Informatika Fakultas Ilmu Komputer Angkatan 2019
5. Henki Bayu Seta, S.Kom, MTI. selaku Dosen Pembimbing 1.
6. Neny Rosmawarni, M.Kom selaku Dosen Pembimbing 2.

Semoga Tugas Akhir ini bermanfaat bagi pihak yang membutuhkan dan bagi kemajuan ilmu pengetahuan.

Jakarta, 22 Juli 2026

Penulis

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

Judul : Analisis Performa dan Keamanan Algoritma HMAC-SHA-2
dan KMAC-SHA-3 dalam Time-based One Time Password
pada Two-Factor Authentication

Nama : Sandy Ardanu Bagasraga

NIM : 1910511054

Program Studi : S1 Informatika

Disetujui oleh:

Penguji 1:

Prof. Dr. Ir. Supriyanto, ST., M.Sc., IPM

Penguji 2:

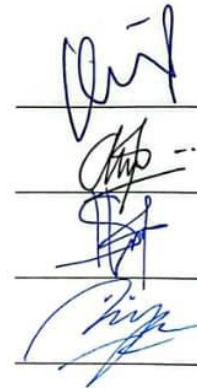
I Wayan Rangga Pinastawa, M.Kom

Dosen Pembimbing 1:

Henki Bayu Seta, S.Kom, MTI.

Dosen Pembimbing 2:

Neny Rosmawarni, M.Kom



Diketahui oleh:

Koordinator Program Studi:

Dr. Ridwan Raafi'udin, S.Kom., M.Kom

NIP. 198805022021211001

Dekan Fakultas Ilmu Komputer:

Prof. Dr. Ir. Supriyanto, ST., M.Sc., IPM

NIP. 197605082003121002



Tanggal Ujian Tugas Akhir:

14 Juli 2026

PERNYATAAN ORISINALITAS

PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya sendiri dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.

Nama : Sandy Ardanu Bagasraga
NIM : 1910511024
Tanggal : 28 Juli 2026

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan saya ini, maka saya bersedia dituntut dan diproses sesuai dengan ketentuan yang berlaku

Jakarta, 28 Juli 2026

Yang Menyatakan



DBB58A0X113032366

Sandy Ardanu Bagasraga

LEMBAR PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

LEMBAR PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademika Universitas Pembangunan Nasional Veteran Jakarta,
saya yang bertanda tangan di bawah ini:

Nama : Sandy Ardanu Bagasraga
NIM : 1910511054
Fakultas : Ilmu Komputer
Program Studi : S-1 Informatika

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada
Universitas Pembangunan Nasional Veteran Jakarta Hak Bebas Royalti Non
eksklusif (*Non - exclusive Royalty Free Right*) atas skripsi saya yang berjudul:

**Analisis Performa dan Keamanan Algoritma HMAC-SHA-2 dan KMAC-
SHA-3 dalam Time-based One Time Password pada Two-Factor
Authentication**

Dengan Hak Bebas Royalti ini Universitas Pembangunan Nasional Veteran Jakarta
berhak menyimpan, mengalih media/memformatkan, mengelola dalam bentuk
pangkalan data (basis data), merawat dan mempublikasikan skripsi saya selama
tetap mencantumkan nama saya sebagai penulis dan sebagai pemilik Hak Cipta.
Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di: Jakarta
Pada tanggal: 28 Juli 2026
Yang Menyatakan



Sandy Ardanu Bagasraga

ABSTRAK

Penggunaan kata sandi tunggal semakin tidak memadai dalam menghadapi ancaman serangan identitas digital, sehingga penerapan *Two-Factor Authentication* (2FA) berbasis *Time-based One Time Password* (TOTP) menjadi standar keamanan yang krusial. Namun, efektivitas sistem TOTP saat ini terhalang oleh penggunaan standar lama HMAC-SHA-2 (seperti HMAC-SHA-256 dan HMAC-SHA-512) yang memanfaatkan arsitektur Merkle-Damgård. Arsitektur ini memerlukan pemrosesan fungsi *hash* secara ganda (*nested double-pass*) untuk mencegah serangan *length-extension*, sehingga menimbulkan beban komputasi yang tinggi dan penggunaan memori yang kurang efisien. Penelitian ini bertujuan untuk menganalisis, membandingkan, serta mengevaluasi performa dan tingkat keamanan algoritma HMAC-SHA-2 dengan algoritma KMAC-SHA-3 (KMACXOF128 dan KMACXOF256 berbasis *Sponge Construction* / KECCAK) dalam implementasi TOTP 2FA. Pengujian dilakukan menggunakan pendekatan eksperimental dan analisis matematis. Aspek performa diuji meliputi kecepatan *generate* dan verifikasi kode TOTP (pada skenario 1.000 *secret key* dan 1.000.000 total eksekusi), penggunaan sumber daya (CPU dan RAM), serta *throughput* dan *multi-thread*. Aspek keamanan dievaluasi melalui uji *avalanche effect*, keacakan (*randomness* menggunakan NIST *Statistical Test Suite*), serta analisis matematis pada *collision resistance*, *preimage resistance*, dan *second-preimage resistance*. Hasil analisis dan perbandingan ini diharapkan dapat memberikan referensi akademis serta panduan teknis bagi pengembang sistem keamanan digital dalam melakukan transformasi dan migrasi standar dari HMAC ke KMAC pada skema TOTP masa depan.

Kata Kunci: TOTP, 2FA, HMAC-SHA-2, KMAC-SHA-3, Performa dan Keamanan Kriptografi.

ABSTRACT

The rely on single passwords has become increasingly inadequate in mitigating digital identity threats, making the implementation of Time-based One Time Password (TOTP) within Two-Factor Authentication (2FA) a critical security standard. However, the current effectiveness of TOTP systems is constrained by the legacy standard HMAC-SHA-2 (such as HMAC-SHA-256 and HMAC-SHA-512), which relies on the Merkle-Damgård architecture. This architecture requires a nested double-pass hash execution to prevent length-extension attacks, resulting in high computational overhead and inefficient memory utilization. This study aims to analyze, compare, and evaluate the performance and security strength of HMAC-SHA-2 algorithms against KMAC-SHA-3 algorithms (KMACXOF128 and KMACXOF256 based on Keccak Sponge Construction) in TOTP-2FA implementations. The evaluation was conducted using an experimental approach combined with mathematical analysis. Performance metrics included TOTP code generation and verification latency (tested across 1,000 secret keys and 1,000,000 total executions), resource usage (CPU and RAM), as well as throughput and multi-threading capabilities. Security properties were assessed via avalanche effect testing, bit randomness (using the NIST Statistical Test Suite), and mathematical analysis of collision resistance, preimage resistance, and second-preimage resistance. The comparative findings provide academic reference and technical guidance for digital security developers transitioning from legacy HMAC to modern KMAC standards in future TOTP authentication systems.

Keywords: TOTP, 2FA, HMAC-SHA-2, KMAC-SHA-3, Cryptographic Performance and Security.

DAFTAR ISI

KATA PENGANTAR	ii
ABSTRAK	iii
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR RUMUS	xii
DAFTAR LAMPIRAN	xiv
BAB I. PENDAHULUAN	1
1.1. Pendahuluan	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan dan Manfaat Penelitian	4
1.4.1. Tujuan Penelitian	4
1.4.2. Manfaat Penelitian	4
1.5. Sistematika Penulisan	4
BAB II. TINJAUAN PUSTAKA	6
2.1. Kajian Teoritis	6
2.1.1. <i>Message Authentication Code</i> (MAC)	6
2.1.2. <i>Hash-based Message Authentication Code</i> (HMAC)	7
2.1.3. KECCAK <i>Message Authentication Code Extendable-Output Function</i> (KMACXOF)	9
2.1.4. <i>Multi-Factor Authentication</i> (MFA)	12
2.1.5. Fungsi <i>Hash</i>	15
2.1.6. Python	26
BAB III. METODE PENELITIAN	27
3.1. Metode Penelitian	27
3.1.1. Perencanaan	28

3.1.2.	Analisis.....	28
3.1.3.	Perancangan	28
3.1.4.	Pengujian.....	29
3.1.5.	Analisis dan Evaluasi Hasil.....	29
3.2.	Metode yang Diusulkan	29
3.2.1.	Perencanaan.....	30
3.2.2.	Analisis.....	30
3.2.3.	Perancangan	30
3.2.4.	Pengujian.....	42
3.2.5.	Analisis dan Evaluasi Hasil.....	46
3.3.	Teknik Pengumpulan Data.....	47
3.4.	Metode Analisis	47
BAB IV. HASIL DAN PEMBAHASAN		54
4.1.	Hasil Pengujian Performa	54
4.1.1.	Kecepatan <i>Generate</i> dan <i>Verifikasi</i> Kode TOTP	54
4.1.2.	Penggunaan Sumber Daya (<i>Resources</i>)	59
4.1.3.	<i>Throuput</i> dan <i>Multi-thread</i>	63
4.2.	Hasil Pengujian Keamanan	69
4.2.1.	<i>Avalanche Effect</i>	70
4.2.2.	Tingkat Keacakan (<i>Randomness</i>).....	74
4.2.3.	<i>Collision Resistance</i>	77
4.2.4.	<i>Preimage</i> dan <i>Second-preimage Resistance</i>	84
4.3.	Evaluasi Hasil Pengujian.....	92
4.3.1.	Evaluasi Pengujian Performa	92
4.3.2.	Evaluasi Pengujian Keamanan.....	93
BAB V. KESIMPULAN DAN SARAN.....		95
5.1.	Kesimpulan	95
5.2.	Saran.....	96
DAFTAR PUSTAKA		97
LAMPIRAN		1

DAFTAR GAMBAR

Gambar 2. 1 Proses Otentikasi TOTP	14
Gambar 2. 2 Proses Satu Putaran SHA-2.....	16
Gambar 2. 3 <i>Sponge Structure</i>	21
Gambar 3. 1 Diagram Metode Eksperimental.....	27
Gambar 3. 2 Flowchart Contoh Implementasi Python SHA-256 dalam Pengujian Generate dan Verifikasi Kode TOTP	31
Gambar 3. 3 Flowchart Proses Algoritma HMAC dalam TOTP	32
Gambar 3. 4 Flowchart Implementasi Python Algoritma cSHAKE dalam TOTP	33
Gambar 3. 5 Flowchart Implementasi Python Algoritma KMACXOF dalam TOTP	34
Gambar 3. 6 Contoh Hasil Pengujian Kecepatan Generate dan Verifikasi Kode TOTP Algoritma	42
Gambar 3. 7 Contoh Hasil Pengujian Penggunaan Sumber Daya	42
Gambar 3. 8 Contoh Hasil Pengujian Throughput dan Multi-thread.....	43
Gambar 4. 1 Grafik Batang Rata-Rata Kecepatan Generate dan Verifikasi Kode TOTP	54
Gambar 4. 2 Grafik Perbandingan Rata-Rata Pengujian Penggunaan Sumber Daya (Resources).....	59
Gambar 4. 3 Grafik Perbandingan Hasil Pengujian Throughput dan Multi-thread	63
Gambar 4. 4 Grafik Perbandingan Hasil Pengujian Avalanche Effect	70
Gambar 4. 5 Tabel Rekapitulasi Hasil Pengujian Avalanche Effect	72
Gambar 4. 6 Grafik Persebaran Bit Hasil Pengujian Frequency Test.....	75

DAFTAR TABEL

Tabel 2. 1 Variabel Awal Hash	18
Tabel 2. 2 Konstanta SHA-256	18
Tabel 2. 3 Parameter Algoritma SHAKE128 dan SHAKE256.....	23
Tabel 2. 4 Tabel Perbandingan Hasil Pengujian Penggunaan Sumber Daya.....	61
Tabel 3. 1 Perbedaan Parameter Algoritma SHA-256 dan SHA-512	34
Tabel 3. 2 Variabel Awal Hash	35
Tabel 3. 3 Konstanta SHA-256	36
Tabel 3. 4 Perbedaan Algoritma SHAKE dengan cSHAKE.....	38
Tabel 4. 1 Tabel Rekapitulasi Data Mentah Hasil Pengujian Kecepatan Generate dan Verifikasi Kode TOTP	57
Tabel 4. 2 Tabel Rekapitulasi Hasil Pengujian Throughput dan Multi-thread	65
Tabel 4. 3 Tabel Rekapitulasi Hasil 15 Pengujian NIST STS	76

DAFTAR RUMUS

(2. 1) Rumus Penentuan/ <i>Padding</i> Kunci pada HMAC	7
(2. 2) Rumus <i>Outer Key Pad</i> (Opad) pada HMAC	8
(2. 3) Rumus <i>Inner Key Pad</i> (Ipad) pada HMAC	8
(2. 4) Rumus <i>Outer Key Pad Vector</i> (oKeyPad)	8
(2. 5) Rumus <i>Inner Key Pad Vector</i> (iKeyPad).....	8
(2. 6) Rumus Perhitungan <i>Digest</i> HMAC	8
(2. 7) Rumus Eksekusi Utam/Sintaksis KMACXOF	10
(2. 8) Rumus Pembentukan Variabel Masukan <i>newX</i> pada KMACXOF	10
(2. 9) Rumus Pembentukan Masukan <i>newX</i> KMACXOF128	11
(2. 10) Rumus Kalkulasi Fungsi KMACXOF128 berbasis cSHAKE128	11
(2. 11) Rumus Variabel <i>T</i> (Konstruksi Sponge KECCAK[256] KMACXOF128)	11
(2. 12) Rumus <i>Output</i> KECCAK[256] KMACXOF128	11
(2. 13) Rumus Pembentukan Masukan <i>newX</i> KMACXOF256	12
(2. 14) Rumus Kalkulasi Fungsi KMACXOF128 berbasis cSHAKE256	12
(2. 15) Rumus Ekspansi Pesan SHA-256 (W_t)	17
(2. 16) Rumus Kalkulasi Fungsi SHAKE ($SHAKE_{X,L}$) berbasis cSHAKE	22
(2. 17) Rumus <i>Padding Input</i> Pesan (M') pada SHAKE	24
(2. 18) Rumus Fungsi <i>Padding pad</i> 10^*1	24
(2. 19) Rumus Pemisahan Blok Data pada SHAKE	24
(2. 20) Rumus Inisialisasi <i>Internal State</i> (S) pada SHAKE	24
(2. 21) Rumus Operasi XOR <i>Rate State</i> (S_r) pada SHAKE	24
(2. 22) Rumus Permutasi <i>State</i> Keccak-p	25
(2. 23) Rumus Ekstraksi <i>Output</i> pada <i>Squeeze Phase</i> SHAKE	25
(2. 24) Rumus Kondisi cSHAKE Tanpa Kustomisasi (Ekuivalen SHAKE)	25
(2. 25) Rumus Kondisi cSHAKE dengan Kustomisasi ($N \neq ""\$ \text{ atau } \$S \neq ""$)	26
(3. 1) Rumus Komputasi <i>Temp 1</i> (T_1) Pembuatan <i>Message Digest</i> SHA-256	37
(3. 2) Rumus Komputasi <i>Temp 1</i> (T_2) Pembuatan <i>Message Digest</i> SHA-256	37
(3. 3) Rumus Probabilitas Kolisi Generik (<i>Birthday Paradox</i>)	43
(3. 4) Rumus Estimasi Jumlah Sampel (k) untuk Target Probabilitas Kolisi	44
(3. 5) Rumus Penyederhanaan Batas Keamanan Kolisi ($p = 0,5$).....	44
(3. 6) Rumus Probabilitas Diferensial <i>Single Operation</i> (DP)	44
(3. 7) Rumus Total Probabilitas Diferensial R-Putaran (P_{total})	44
(3. 8) Rumus Pengukuran Jarak <i>Hamming</i> (<i>Hamming Distance</i>).....	44
(3. 9) Rumus Rasio <i>Strict Avalanche Criterion</i> (SAC).....	45
(3. 10) Rumus Uji Statistik <i>Chi-Square</i> (χ^2).....	45
(3. 11) Rumus Probabilitas Pencocokan <i>First Preimage</i>	45

(3. 12) Rumus Ekspektasi Kompleksitas <i>First Preimage Brute-Force</i>	46
(3. 13) Rumus Kompleksitas <i>Second Preimage</i> (Kelsey-Schneier <i>Long Message Attack</i>)	46
(3. 14) Rumus Kompleksitas Serangan <i>Meet-in-the-Middle</i> (MitM).....	46
(3. 17) Rumus Kompleksitas Solusi Aljabar (Gröbner Basis <i>Complexity</i>).....	46
(3. 18) Rumus Operasi Bitwise σ_1 Ekspansi Pesan SHA-256	48
(3. 19) Rumus Operasi Bitwise σ_0 Ekspansi Pesan SHA-256	48
(3. 20) Rumus Operasi <i>Bitwise</i> Σ_1 Komputasi <i>Digest</i> SHA-256	49
(3. 21) Rumus Operasi Logika <i>Ch</i> (<i>Choice</i>) SHA-256	49
(3. 22) Rumus Operasi Bitwise Σ_0 Komputasi <i>Digest</i> SHA-256.....	49
(3. 23) Rumus Operasi Logika <i>Maj</i> (<i>Majority</i>) SHA-256.....	50

DAFTAR LAMPIRAN

Lampiran 1 Pseudocode Python Contoh Implementasi SHA-256 dalam Pengujian Generate dan Verifikasi Kode TOTP.....	1
Lampiran 2 Pseudocode Python Contoh Implementasi HMAC-SHA-256 dalam TOTP	1
Lampiran 3 Pseudocode Python Contoh Implementasi cSHAKE dalam TOTP	2
Lampiran 4 Pseudocode Python Contoh Implementasi KMACXOF dalam TOTP	3
Lampiran 5 Biner Lengkap Hasil Panjang Pesan SHA-256 Setelah Padding	3
Lampiran 6 Operasi Bitwise Ekspansi Pesan W_{16}	4
Lampiran 7 Operasi Bitwise dari Operasi Σ_1 , Ch, dan T1	4
Lampiran 8 Nilai Hash Akhir SHA256.....	5
Lampiran 9 Biner State Hasil Operasi Pemetaan cSHAKE128.....	6
Lampiran 10 Kode Python Pengujian Performa dan Keamanan serta Data Mentah Hasil Pengujian.....	6
Lampiran 11 Grafik Batang Perbandingan Kecepatan Generate dan Verifikasi Kode TOTP per Secret Key	7
Lampiran 12 Grafik Perbandingan Penggunaan Sumber Daya per Secret Key.....	8
Lampiran 13 Grafik Perbandingan Lengkap Persentase Avalanche Effect	10