BAB V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian, analisis, perbandingan dan evaluasi performa dan kemananan dari algoritma SHA-256, SHA-512, HMAC-SHA-256, HMAC-SHA-512, SHAKE128, SHAKE256, KMACXOF128 dan KMACXOF256 dalam implementasi *Time-based One Time Password* (TOTP) pada *Two-Factor Authentication* (2FA), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Analisis performa dan keamanan algoritma HMAC-SHA-2 serta KMAC-SHA-3 berhasil dilakukan melalui pengujian berbasis *Python* dengan mengukur kecepatan *generate* dan verifikasi kode, penggunaan sumber daya (CPU dan RAM), serta throughput multi-threading terhadap 1.000.000 kode TOTP. Hasil analisis arsitektur menunjukkan bahwa struktur *Sponge Construction* (KECCAK) pada KMAC-SHA-3 memberikan keunggulan keamanan berupa perlindungan alami dari *length-extension attack* serta mendukung fitur *domain separation* menggunakan parameter *Function-Name* (*N*) dan *Customization String* (*S*).
2. Hasil pengujian menunjukkan bahwa KMACXOF (KMACXOF128 dan KMACXOF256) lebih unggul dalam kecepatan total (~0,059 detik) dan efisiensi waktu CPU (~51–57 detik pada variasi 8 dan 10 digit) dibandingkan HMAC-SHA-2. Meski demikian, HMAC-SHA-2 sedikit lebih hemat RAM (~19,9–20,3 MB) dibandingkan KMACXOF (~22,3 MB) dan mendominasi *throughput multi-threaded* (14.351–17.657 RPS) akibat adanya *thread contention* pada pustaka KMACXOF dalam pemrosesan paralel (4.500–5.500 RPS). Dari segi keamanan, semua algoritma yang diuji menunjukkan tingkat keamanan terhadap *collision, preimage* dan *second-preimage* yang cukup baik, di mana dibutuhkan 2^{128} operasi untuk SHA-256/HMAC-SHA-256/cSHAKE (varian 256-bit)/KMACXOF (varian 256-bit) dan 2^{256} untuk SHA-512/HMAC-SHA-512/cSHAKE (varian 512-bit)/KMACXOF (varian 512-bit).
3. Algoritma KMAC-SHA-3 terbukti efisien dan sangat layak diterapkan sebagai alternatif pengganti HMAC-SHA-2 pada sistem TOTP 2FA, khususnya untuk lingkungan pemrosesan *single-threaded* atau perangkat dengan keterbatasan komputasi. Namun, untuk arsitektur *server* dengan lalu lintas transaksi sangat tinggi yang mengandalkan pemrosesan konkurensi *multi-threaded*, HMAC-SHA-256 masih menawarkan performa *throughput* pemrosesan paralel yang lebih optimal.

5.2. Saran

Berdasarkan hasil penelitian dan keterbatasan yang ada, peneliti memberikan beberapa saran untuk pengembangan penelitian selanjutnya sebagai berikut:

1. Disarankan untuk melakukan pengujian ulang menggunakan bahasa pemrograman *low-level* (seperti C, Rust, atau Go) serta pustaka kriptografi *native* (seperti OpenSSL 3.x). Langkah ini penting untuk mengonfirmasi apakah penurunan *throughput multi-threading* pada KMAC-SHA-3 murni disebabkan oleh *thread contention* atau *lock overhead* pada pustaka Python yang digunakan, atau merupakan karakteristik internal algoritma.
2. Penelitian selanjutnya dapat mengimplementasikan dan menguji TOTP berbasis KMAC-SHA-3 secara langsung pada perangkat keras dengan sumber daya terbatas, seperti *microcontroller*, perangkat IoT, maupun aplikasi *mobile.*