Jurnal Quancom
Vol.3, No.1, Juni 2025, pp. 13~17
e-ISSN: 2988-0998
P-ISSN: 2988-1005
DOI: https://doi.org/10.62375/jqc.v3i1.431

Analisis Terhadap Keamanan Password Menggunakan Hash SHA-256

Vivaldi Simangunsong [1]*, Yen Rylin Hutasoit[2], Desrico Siallagan [3s]
[1,2,3] Teknik Rekayasa Perangkat Lunak, Institut Teknologi Del
[1]if422042@students.del.ac.id, [2]if422055@students.del.ac.id, [3]if422025@students.del.ac.id

Article Info

Article history:

Received 4 Desember 2024
Revised 24 Desember 2024
Accepted 25 Desember 2024

Keyword:

Keamanan, Hashing, SHA-256, Algoritma Hash, Keamanan Data, Kriptografi

ABSTRACT

In today's digital era, password security is a critical aspect of protecting user data from various threats. This study focuses on analyzing the SHA-256 hashing algorithm, widely recognized for its ability to generate unique and unpredictable hash values. By simulating several password patterns, the research evaluates the algorithm's effectiveness against attacks such as brute force and rainbow table. The findings reveal that SHA-256 is reliable in safeguarding passwords against collision-based threats. However, the capabilities of modern hardware to test billions of combinations per second make password length and complexity key factors in mitigating attack risks. This study highlights the strengths and limitations of SHA-256 in password security systems. It is expected that the findings will assist developers in implementing this algorithm to create more secure systems.

This is an open access article under the CC Attribution 4.0 license.

PENDAHULUAN

Keamanan data merupakan salah satu elemen terpenting dalam era digital, terutama dalam melindungi informasi pribadi dan akses terhadap sistem berbasis internet. Kata sandi, sebagai metode autentikasi yang paling umum digunakan, menjadi fokus utama dalam berbagai penelitian keamanan informasi. Namun, tantangan utama yang muncul adalah bagaimana memastikan kata sandi tetap aman meskipun data yang mengandung hash kata sandi bocor ke public [1].

Hashing adalah teknik kriptografi yang mengubah data menjadi nilai tetap dengan panjang tertentu menggunakan fungsi matematis. Proses ini tidak dapat dibalikkan, yang berarti data asli tidak dapat direkonstruksi dari nilai hash yang dihasilkan[2]. Salah satu algoritma hashing yang banyak digunakan adalah SHA-256, yang merupakan bagian dari keluarga Secure Hash Algorithm (SHA) dan memiliki reputasi tinggi dalam menghasilkan nilai hash yang unik dan tahan terhadap kolisi[3].

SHA-256 dirancang untuk memberikan keamanan tingkat tinggi dengan panjang output 256-bit. Dalam implementasinya, algoritma ini memecah data menjadi blok-blok 512-bit, kemudian menjalani serangkaian transformasi matematis yang memastikan bahwa nilai hash yang dihasilkan sangat sulit ditebak[4]. Dalam konteks keamanan kata sandi, SHA-256 memberikan proteksi terhadap serangan yang mencoba menebak kata sandi melalui eksploitasi data hash[5].

Meski SHA-256 sangat efektif melawan serangan berbasis kolisi, serangan *brute force* tetap menjadi ancaman yang signifikan, terutama dengan perkembangan perangkat keras modern yang memungkinkan miliaran tebakan per detik[6]. Penelitian menunjukkan bahwa panjang dan kompleksitas kata sandi memainkan peran penting dalam ketahanan terhadap serangan ini[7]. Kata sandi yang lebih panjang dengan kombinasi karakter yang kompleks secara eksponensial meningkatkan waktu yang dibutuhkan untuk memecahkannya.

Dalam penelitian ini, analisis terhadap keamanan SHA-256 dilakukan dengan mensimulasikan beberapa skenario penggunaan kata sandi. Pendekatan ini memberikan wawasan langsung mengenai tingkat resistansi algoritma terhadap ancaman dunia nyata[8].

Beberapa penelitian sebelumnya telah menyoroti kinerja SHA-256 dalam berbagai aplikasi keamanan[9]. membandingkan kecepatan dan keamanan SHA-256 dengan algoritma SHA-1, menunjukkan bahwa SHA-256 lebih unggul dalam menghadapi serangan berbasis. Penelitian lain oleh [10] menguji kinerja algoritma SHA-256 pada perangkat keras yang berbeda, mengungkapkan

bahwa kecepatan hashing bervariasi tergantung pada spesifikasi sistem. Selain itu, [11] mengevaluasi efektivitas hashing untuk melindungi sistem autentikasi berbasis kata sandi dari berbagai jenis serangan.

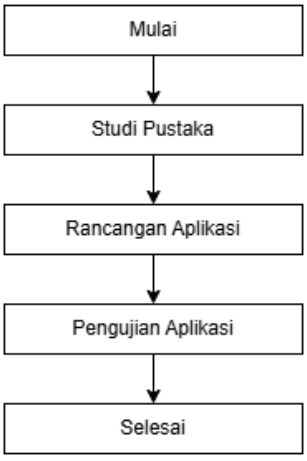
Penelitian oleh Specops Software (2024) memberikan perspektif baru mengenai waktu yang dibutuhkan untuk memecahkan hash SHA-256 berdasarkan panjang dan kompleksitas kata sandi. Hasil penelitian ini relevan dengan penelitian yang dilakukan, mengingat simulasi yang dilakukan pada proyek ini juga mengukur ketahanan kata sandi terhadap serangan berbasis kekuatan komputasi.

Penelitian ini berkontribusi dalam mengevaluasi efektivitas algoritma SHA-256 dalam melindungi kata sandi. Fokus utama adalah pada analisis tingkat keamanan yang diberikan algoritma ini terhadap berbagai jenis pola kata sandi. Dengan pendekatan yang disesuaikan dengan implementasi nyata, hasil penelitian ini diharapkan dapat memberikan panduan untuk mengidentifikasi kekuatan dan kelemahan algoritma SHA-256 dalam konteks keamanan sistem autentikasi berbasis kata sandi.

Pendahuluan ini memberikan latar belakang, tujuan, dan kontribusi dari penelitian. Bab berikutnya akan menjelaskan metodologi yang digunakan, termasuk teknik pengumpulan data dan simulasi yang dilakukan. Hasil dan pembahasan akan menguraikan temuan utama terkait kinerja SHA-256 dalam menjaga keamanan kata sandi. Akhirnya, kesimpulan akan merangkum kontribusi penelitian dan rekomendasi untuk pengembangan lebih lanjut.

METODE

Metode penelitian yang dilakukan dalam penelitian ini meliputi studi pustaka, konsultasi, rancangan aplikasi dan pengujian aplikasi. Model kerangka kerja yang akan digunakan dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Kerangka kerja penelitian

Keterangan alur kerangka kerja penelitian Gambar 1, dengan tahapan:

Tahap awal penelitian dimulai dengan mengumpulkan data dari literatur yang relevan, seperti jurnal ilmiah, laporan teknis, dan standar keamanan terkait hashing, khususnya algoritma SHA-256. Studi ini bertujuan untuk memahami konsep hashing, jenis ancaman keamanan, serta teknik tambahan seperti *salting* dan iterasi hashing.

Dataset password yang digunakan dalam penelitian ini mencakup variasi password lemah, sedang, dan kuat. Dataset tersebut dikumpulkan dari sumber yang aman atau dibuat secara sintetis, dengan referensi dari penelitian sebelumnya mengenai penggunaan password lemah di Indonesia. Dataset ini penting untuk menguji ketahanan SHA-256 dalam berbagai kondisi.

Tahap selanjutnya adalah rancangan aplikasi, yaitu merancang aplikasi dengan metode *caesar cipher* yang dipergunakan pada Aplikasi. Aplikasi keamanan data ini dirancang untuk meningkatkan keamanan data dan informasi rahasia milik pribadi maupun kelompok atau organisasi dengan media web. Perancangan dan pembuatan aplikasi ini dirancang melibatkan implementasi hashing password menggunakan algoritma SHA-256 dengan Python. Proses ini melibatkan:

Pembuatan dataset password dengan berbagai tingkat kompleksitas.

Penambahan fitur *salting* dan iterasi hashing untuk meningkatkan keamanan.

Pengujian fungsi hash untuk memastikan nilai hash bersifat unik dan aman.

Tahap terakhir adalah pengujian aplikasi, dalam pengujian aplikasi keamanan data ini akan diujicobakan dengan menggunakan pesan rahasia berupa *plaintext*, yang diharapkan menghasilkan keluaran berupa *ciphertext* yang dapat digunakan dengan jumlah 256 karakter. Uji coba kebenaran metode tersebut juga akan dilakukan menggunakan rumus enkripsi dan dekripsi *Caesar cipher* dengan jumlah 256 karakter.

Pengujian ini dilakukan juga untuk mengevaluasi keamanan SHA-256 dalam menghadapi beberapa jenis serangan, seperti:

Brute Force Attack: Menguji waktu yang dibutuhkan untuk menebak password menggunakan berbagai kombinasi.

Rainbow Table Attack: Mengukur sejauh mana serangan berbasis tabel pra-hash dapat berhasil.

Dictionary Attack: Menganalisis kerentanan terhadap daftar password umum.

HASIL DAN PEMBAHASAN

A. Pendahuluan

Pada bab ini, akan dibahas mengenai rancangan aplikasi yang digunakan untuk menganalisis keamanan password dengan menggunakan algoritma hash SHA-256. Aplikasi ini dirancang untuk memberikan gambaran kepada pengguna mengenai seberapa aman sebuah password berdasarkan beberapa faktor, seperti panjang password, kompleksitas karakter yang digunakan, serta ketahanan password tersebut terhadap serangan brute force. Aplikasi ini juga bertujuan untuk membantu pengguna dalam memilih password yang kuat dan memperkenalkan pentingnya hashing dalam pengamanan data pribadi.

B. Tujuan Aplikasi

Tujuan dari aplikasi ini adalah untuk menganalisis dan memberikan informasi terkait keamanan password, dengan menggunakan metode hashing SHA-256 sebagai dasar perlindungan data. Aplikasi ini memiliki beberapa fungsi utama yang meliputi:

Evaluasi Kekuatan Password

Aplikasi akan menilai kekuatan password berdasarkan kriteria seperti panjang, variasi karakter (huruf besar, huruf kecil, angka, dan simbol), serta kerentanannya terhadap serangan brute force.

Estimasi Waktu Brute Force

Menghitung waktu yang dibutuhkan untuk mencoba semua kemungkinan kombinasi password menggunakan teknik brute force, yang akan dihitung berdasarkan panjang password dan jumlah karakter yang digunakan.

Proses Hashing dengan SHA-256

Menggunakan algoritma SHA-256 untuk mengubah password menjadi nilai hash yang tidak dapat dibalikkan, yang dapat digunakan untuk mengamankan password dalam sistem autentikasi.

C. Deskripsi Sistem

Sistem aplikasi ini akan dibangun dengan menggunakan framework Flask untuk sisi backend dan HTML, CSS, JavaScript untuk sisi frontend. Pengguna dapat mengakses aplikasi melalui web browser dan memasukkan password untuk diuji. Berikut adalah gambaran alur kerja aplikasi:

Frontend:

Antarmuka pengguna yang intuitif memungkinkan pengguna untuk memasukkan password. Menampilkan hasil analisis kekuatan password, estimasi waktu brute force, dan nilai hash SHA-256. Tampilan yang responsif, dengan grafik dan indikator yang memperlihatkan tingkat keamanan password berdasarkan kriteria yang telah disebutkan.

Backend (Flask):

Route untuk halaman utama (/), menyediakan form input bagi pengguna untuk memasukkan password.

API untuk analisis password (/api/analyze), menghitung kekuatan password, estimasi waktu brute force, dan menghasilkan nilai hash SHA-256.

Perhitungan keamanan, sistem menghitung waktu yang diperlukan untuk memecahkan password menggunakan brute force dan memberikan rekomendasi keamanan berdasarkan hasil tersebut.

Algoritma:

SHA-256 Hashing, password yang dimasukkan akan diubah menjadi nilai hash SHA-256 yang aman.

D. Rancangan UI/UX

Antarmuka pengguna dirancang dengan prinsip simplicity dan usability untuk memastikan pengguna dapat dengan mudah memahami dan mengoperasikan aplikasi. Elemen-elemen utama dalam UI antara lain:

Form Input Password:

Pengguna dapat memasukkan password pada kolom yang disediakan.

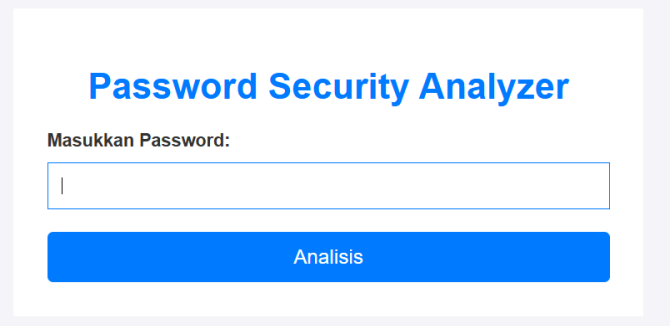
Tombol untuk mengirim password ke server untuk dianalisis.

Hasil Analisis. Tampil informasi tentang kekuatan password (lemah, sedang, kuat), estimasi waktu brute force untuk memecahkan password, nilai hash SHA-256 dari password yang dimasukkan.

Contoh tampilan antarmuka:

Halaman Input

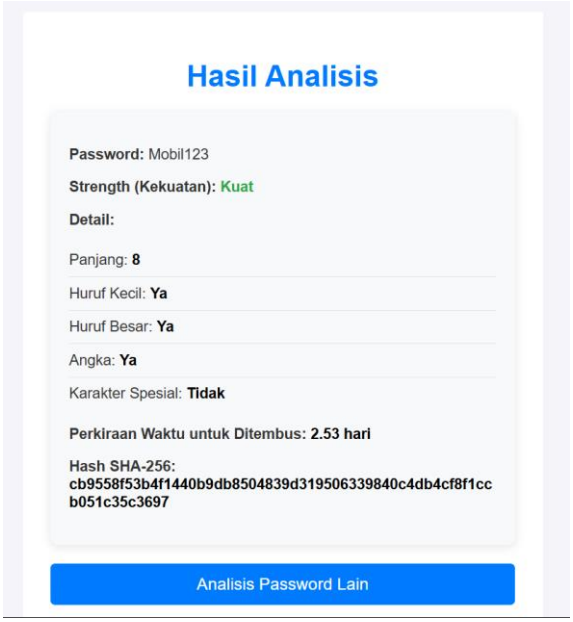
Memuat form untuk memasukkan password dan menampilkan tombol "Analisis".



Gambar 2. Tampilan halaman input

Halaman Hasil

Menampilkan hasil analisis dalam format teks.



Gambar 3. Tampilan halaman hasil

E. Teknologi yang Digunakan

Aplikasi ini menggunakan beberapa teknologi untuk memastikan fungsionalitas dan performa yang optimal. Teknologi yang digunakan antara lain:
Flask, framework Python yang digunakan untuk membangun backend aplikasi.
HTML/CSS/JavaScript, untuk membangun antarmuka pengguna.
Web Cryptography API, untuk melakukan hashing SHA-256 di sisi klien, menjaga privasi data pengguna.
Python Libraries, hashlib untuk menghasilkan hash SHA-256.

F. Alur Kerja Aplikasi

Pengguna memasukkan password pada form input lalu sistem melakukan validasi terhadap password dan mengkategorikan kekuatan password berdasarkan panjang dan kompleksitas karakter yang ada selanjutnya sistem menghitung estimasi waktu brute force dengan memperkirakan jumlah kombinasi yang harus dicoba dan hasil analisis ditampilkan di UI, termasuk waktu estimasi brute force dan nilai hash SHA-256.

G. Proses Hashing SHA-256

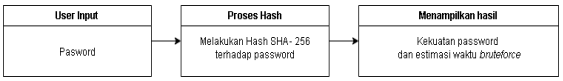
Pada bagian ini, dijelaskan bagaimana SHA-256 digunakan untuk mengubah password menjadi format hash yang tidak dapat dibalikkan. Proses hashing dilakukan menggunakan pustaka hashlib pada backend, di mana password yang dimasukkan oleh pengguna diubah menjadi string panjang 64 karakter. Nilai hash ini digunakan untuk penyimpanan password dalam sistem yang aman.
Langkah-langkah proses hashing:
Mengambil password input dari pengguna.
Menggunakan algoritma SHA-256 untuk menghasilkan nilai hash dari password.
Menampilkan hasil hash tersebut kepada pengguna.

H. Estimasi Waktu Brute Force

Aplikasi menghitung estimasi waktu brute force berdasarkan dua faktor utama:
Panjang Password: Semakin panjang password, semakin besar jumlah kombinasi yang harus dicoba.
Kompleksitas Password: Password yang mengandung huruf besar, kecil, angka, dan simbol akan memiliki lebih banyak kombinasi yang mungkin, sehingga memperpanjang waktu brute force.

I. Diagram Alur Aplikasi

Untuk memberikan gambaran yang lebih jelas mengenai alur kerja aplikasi, berikut adalah diagram alur (flowchart) yang menggambarkan proses dari awal hingga akhir.



Gambar 4. Alur aplikasi

J. Keamanan Aplikasi

Untuk memastikan keamanan dan privasi pengguna:
Hashing di sisi klien: Password yang dimasukkan akan di-hash menggunakan SHA-256 di sisi klien sebelum dikirim ke server.

- Tidak menyimpan password asli, hanya nilai hash yang disimpan, menjaga kerahasiaan password pengguna.
- Keamanan HTTPS, server aplikasi akan menggunakan HTTPS untuk mengenkripsi komunikasi antara pengguna dan server

SIMPULAN

Berdasarkan analisis yang dilakukan, algoritma SHA-256 merupakan solusi yang andal untuk mengamankan kata sandi, terutama jika digunakan bersama teknik pengacakan seperti salt dan iterasi berulang. Penggunaan salt yang unik untuk setiap pengguna mampu secara signifikan mengurangi risiko serangan berbasis tabel pelangi. Selain itu, iterasi berulang memperlambat proses brute force, sehingga meningkatkan keamanan sistem secara keseluruhan. Meski algoritma SHA-256 memiliki kelemahan, terutama terhadap serangan dengan perangkat keras canggih, penerapan praktik terbaik seperti pengelolaan salt dan iterasi berulang dapat memitigasi kelemahan tersebut.

Studi ini juga menekankan pentingnya pendekatan keamanan berlapis, termasuk autentikasi multifaktor dan pembatasan upaya login yang gagal, untuk memperkuat perlindungan sistem autentikasi. Sebagai langkah ke depan,

pengembang sistem dianjurkan untuk terus memantau perkembangan ancaman keamanan dan memperbarui teknik perlindungan mereka guna menghadapi tantangan baru yang muncul di era digital. Implementasi hasil penelitian ini diharapkan dapat meningkatkan kepercayaan pengguna terhadap sistem autentikasi berbasis kata sandi, sekaligus memberikan perlindungan yang lebih baik terhadap informasi sensitif mereka.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Institut Teknologi Del atas dukungan yang diberikan untuk penelitian ini. Bantuan yang kami terima sangat berarti bagi kelancaran dan keberhasilan proyek ini.

Kami juga berterima kasih kepada seluruh pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, sehingga riset ini dapat terselesaikan dengan baik. Semoga kerja sama ini dapat terus terjalin dan memberikan kontribusi positif bagi pengembangan ilmu pengetahuan dan teknologi.

DAFTAR PUSTAKA

[1] S. M. Metev & V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of Web authentication schemes," IEEE Symposium on Security and Privacy (SP), pp. 553–567, 2012, doi: 10.1109/SP.2012.44.

[3] Specops Software, "How well does SHA256 protect against modern password cracking?," 2024. [Online]. Available: https://specopssoft.com/.

[4] W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. Pearson, 2021.

[5] M. Syafriadi, "Analisis kecepatan dan keamanan algoritma secure hash algorithm 256 (SHA-256) untuk otentikasi teks," IPB Repository, 2006. [Online]. Available: http://repository.ipb.ac.id/handle/123456789/48487.

[6] National Institute of Standards and Technology (NIST), "Secure hash standard (SHS)," FIPS PUB 180-4, 2015.

[7] M. Rouse, "What is SHA-256?," TechTarget, 2020. [Online]. Available: https://www.techtarget.com/.

[8] Bonsjak et al., "Perbandingan performa algoritma SHA-256 pada perangkat keras yang berbeda," 2018.

[9] P. A. Grassi et al., "Digital identity guidelines: Authentication and lifecycle management," NIST Special Publication 800-63B, 2017, doi: 10.6028/NIST.SP.800-63B.

[10] Specops Software, "Analisis waktu pemecahan SHA-256 dengan berbagai panjang kata sandi," 2024.