ISSN : 2962 – 5688
Website :
jurnal.umj.ac.id/index.php/TripleA
Email : triplea@umj.ac.id

TRIPLE A
Jurnal Pendidikan Teknologi Informasi

ALGORITME HASHING SHA-512 PADA SISTEM HALAMAN SIGN UP JAVA

Muhammad Anum Fadhillah, Latief Mulyarahim, Kayla Nadira
Politeknik Negeri Jakarta, Fakultas Teknik Informatika dan Komputer, Program Studi Teknik Multimedia dan Jaringan, Universitas Indonesia, Jl. Prof. DR. G.A. Siwabessy, Kukusan, Kecamatan Beji, Kota Depok, Jawa Barat 16425

anumfadhillah@gmail.com

latiefmr06@gmail.com

kayla.nadirah37@gmail.com

Abstract

Secure Hash Algorithm (SHA) plays a vital role in many informational security applications. It was first publicized in 2001 by the National Security Agency in order to secure data from data theft. SHA algorithm is reliable and effective when implemented on legacy systems, the research will focus on performance analysis and system integration. The implementation of this research uses the SHA-512 algorithm on an account registration system based on the java programming language. Where later the results of user password data will be stored encrypted using the SHA-512 algorithm. The use of SHA algorithm integration in the legacy system can ensure information security without reducing performance.

Keywords: *Algoritme SHA, Preimage Attack, Collision Attack, Length Extension Attack*

Abstrak

Secure Hash Algorithm (SHA) memiliki peranan yang vital dalam banyak aplikasi keamanan informasional. Pertama kali dipublikasukan pada tahun 2001 oleh National Security Agency Agar yang digunakan untuk mengamankan data dari pencurian data. Algoritma SHA dapat diandalkan dan efektif ketika diimplementasikan pada sistem warisan (legacy), penelitian akan berfokus pada analisis kinerja dan integrasi sistem. Pengimplementasian penelitian ini menggunakan algoritme SHA-512 pada sistem pendaftaran akun yang berbasis bahasa pemrograman java. Di mana nantinya hasil data sandi pengguna akan tersimpan dengan terenkripsi menggunakan algoritme SHA-512. Penggunaanintegrasi algoritma SHA pada sistem warisan dapat memastikan keamanan informasi tanpa mengurangi performa.

Kata kunci: *Algoritme SHA, Serangan Preimage, Serangan Tabrakan, Serangan Ekstensi Panjang*

TRIPLE A: Jurnal Pendidikan Teknologi Informasi. Vol. 2 (1), pp: 27 - 34

PENDAHULUAN

Keamanan informasi menjadi salah satu aspek yang sangat penting dalam pengembangan sistem dan aplikasi. Salah satu algoritme kriptografi yang luas digunakan untuk menjaga keamanan data adalah SHA (Secure Hash Algorithm). Algoritme ini telah menjadi landasan dalam banyak protokol keamanan, seperti protokol SSL, pembuatan tanda tangan digital, dan verifikasi integritas data.

Penelitian ini bertujuan untuk mengimplementasikan algoritma hashing SHA-512 pada sistem halaman sign up menggunakan bahasa pemrograman Java. Halaman sign up merupakan bagian yang krusial dalam sebuah aplikasi yang memungkinkan pengguna untuk mendaftar dan membuat akun pengguna baru.

Dalam jurnal ini akan merancang dan mengimplementasikan fungsi hashing SHA-512 untuk mengamankan data pengguna, terutama kata sandi (password), pada halaman sign up. Implementasi ini akan melibatkan penggunaan paket java.security yang menyediakan fungsi hash dari algoritma SHA-512.

Dengan mengimplementasikan algoritma hashing SHA-512 pada sistem halaman sign up, diharapkan dapat meningkatkan keamanan data pengguna dan mencegah akses yang tidak sah ke akun pengguna. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem keamanan yang lebih baik dan dapat diandalkan dalam pengembangan aplikasi yang melibatkan proses pendaftaran pengguna.

1. Kriptografi

   Kriptografi berasal dari bahasa Yunani, yaitu “kryptos” dan “graphein”. Kryptos berarti tersembunyi atau rahasia, sedangkan graphein memiliki arti menulis.ilmu kriptografi atau bisa juga disebut kriptologi disebut juga dengan sandisastra. Tujuan dari ilmu kriptografi adalah melakukan berbagai upaya komunikasi antar individu atau kelompok secara aman tanpa digannggu oleh pihak-pihak yang tidak diinginkan. Kriptografi memiliki empat tujuan utama :

   a. Plaintext

      Plaintext dapat berupa teks, kode biner, atau gambar asli sebelum dikonversi menjadi format yang tidak dapat dibaca oleh siapa pun kecuali mereka yang memiliki key untuk membukanya. Ini adalah bentuk data yang dapat dibaca dan dimengerti oleh manusia tanpa melalui proses enkripsi.

   b. Ciphertext

      Hasil dari proses enkripsi yang mengubahplaintext menjadi bentuk yang tidak dapat dibaca atau dimengerti. Ini adalah hasil akhir dari enkripsi dan hanya dapat diubah kembali menjadi 2 plaintext menggunakan proses dekripsi yang tepat dengan menggunakan kunci yang sesuai.

   c. Enkripsi

      Proses mengubah plaintext menjadi ciphertext menggunakan algoritme enkripsi tertentu. Ini melibatkan penggunaan kunci enkripsi yang relevan. Tujuan utama enkripsi adalah untukmelindungi kerahasiaan dan keamanan datadengan mengubahnya menjadi bentuk yang tidak dapat dimengerti oleh pihak yang tidak berwenang.

d. Dekripsi
Proses kebalikan dari enkripsi, yaitu mengubah ciphertext kembali menjadi plaintext menggunakan algoritme dekripsi dan kunci yang sesuai. Hanya penerima yang memiliki kunci dekripsi yang tepat yang dapat mendekripsi ciphertext dan mengembalikannya ke bentuk asli plaintext.

e. Cipher
Cipher merujuk pada algoritme yang digunakan dalam proses enkripsi dan dekripsi. Algoritmecipher mengatur aturan-aturan dan langkah-langkah yang diperlukan untuk mengubah plaintext menjadi ciphertext dan sebaliknya. Beberapa contoh algoritme cipher termasuk AES (Advanced Encryption Standard), RSA, dan DES (Data Encryption Standard).

f. Kunci
Informasi rahasia yang digunakan dalam proses enkripsi dan dekripsi. Kunci berfungsi sebagai "kunci" yang mengontrol algoritme cipher dan memungkinkan untuk mengubah plaintext menjadi ciphertext dan sebaliknya. Kunci yang tepat harus digunakan untuk melakukan dekripsi yang berhasil.

2. SHA (Secure Hash Algorithms)
Secure Hashing Algorithm atau SHA adalah fungsi kriptografi yang dibuat khusus oleh penyedia otoritas keamanan internet dengan tujuan menjaga keamanan data. Di mana, SHA bekerja dengan melakukan transformasi data yang menggunakan fungsi hash.Fungsi hash sendiri arah sehingga tidak bisa diubah ke dalam nilai hash secara masing-masing data karena tergantung pada tingkat bit enkripsi yang digunakan. Selain itu, fungsi hash bisa menghasilkan fungsi acak dengan tidak terlihat aslinya. SHA sendiri memiliki jenis yang sering digunakan seperti SHA-1, SHA-2, serta SHA-256. Di mana, masing-masing SHA memiliki tingkat enkripsi berbeda-beda berdasarkan tingkat kerentanan yang berbeda pula.

a) SHA-1
Jenis SHA-1 dikembangkan pada tahun 1993 oleh lembaga standar pemerintah asal Amerika Serikat alias National Institute of Standard and Technology. SHA 1 akan menghasilkan fungsi hash 160 bit menggunakan panjang kurang dari 2/64 bitsebagai standar keamanan yang masih rendah.

b) SHA-256
SHA-256 adalah algoritme hash yang menghasilkan nilai hash dengan panjang 256-bit atau 32 byte. Algoritme ini mengambil pesan masukan apa pun, baik itu pesan teks, data biner, atau file, dan menghasilkan nilai hash yang unik untuk pesan tersebut. SHA-256 menggunakan transformasi matematis yang kompleks, termasuk operasi bitwise, permutasi, dan fungsi kompresi untuk menghasilkan nilai hash yang tahan terhadap perubahan kecil pada pesan masukan. Nilai hash SHA-256 biasanya ditampilkan sebagai serangkaian angka heksadesimal.

c) SHA-512
SHA-512 adalah varian SHA-2 yang menghasilkan nilai hash dengan panjang 512-bit atau 64 byte.

Algoritme ini menggunakan proses yang serupa dengan SHA-256, tetapi dengan ukuran blok dan transformasi yang lebih besar. SHA-512 menggunakan blok data dengan ukuran 1024-bit, dua kali lebih besar dari SHA-256. Hal ini memberikan tingkat keamanan yang lebih tinggi dan resistensi terhadap serangan kriptografis yang lebih canggih. Nilai hash SHA-512 juga ditampilkan dalam bentuk serangkaian angka heksadesimal. Dalam implementasi SHA, terdapat beberapa jenis serangan yang perlu diperhatikan untuk menjaga keamanan sistem.

Algoritme	Panjang Pesan (bit)	Ukuran Blok (dalam bit)	Ukuran *Word*(dalam bit)	Ukuran *Message Diget* (bit)	*Security* (bit)
SHA-1	$<2_{64}$	512	32	160	80
SHA-256	$<2_{64}$	512	32	256	128
SHA-512	$<2_{128}$	1024	64	512	256

Tabel 1. Perbedaan Variasi Algoritme SHA

3. Pemrograman Java

Java adalah bahasa pemrograman yang dikembangkan oleh James Gosling, Patrick Naughton, Chris Warth, Ed Frank, dan Mike Sheridan di Sun Microsystems (sekarang bagian dari Oracle Corporation) pada tahun 1995. Awalnya, Java dirancang untuk digunakan dalam perangkat elektronikkonsumen, terutama untuk mengoperasikan peralatan rumah tangga yang terhubung ke jaringan, yang dikenal sebagai "Java Ring". Namun, Java kemudian meluas penggunaannya dan menjadi populer dalam pengembangan perangkat lunak karena berbagai keunggulannya. Java merupakan Bahasa pemrograman berbasis OOP yang mendukung konsep seperti enkapsulasi, pewarisan, dan polimorfisme. Pendekatan ini memungkinkan pengembang untuk merancang dan membangun aplikasi yang lebih mudah dipelihara, diubah, dan diperluas.

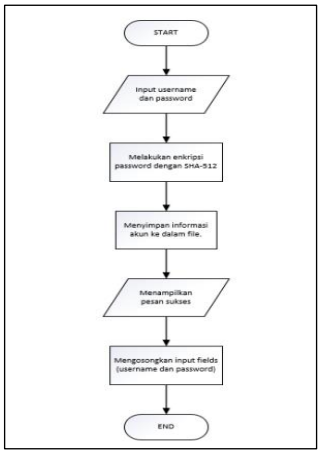
METODE PENELITIAN

Pada penelitian ini dibuat beberapa metode yaitu sebagai berikut:

1. **Metode sha512**
   Metode ini digunakan untuk membuat objek dan menginisialisasi komponen GUI.
2. **Metode *encryptSHA512(String input)***
   Pada metode tersebut dilakukan proses pengenkripsian input menggunakan algoritma SHA512. Metode ini mengembalikan hasil enkripsi dalam bentuk string.
3. **Metode *store Account Info (String username, String password)***
   Metode ini digunakan untuk menyimpan informasi akun pengguna ke dalam file teks. Metode ini menerima dua parameter, yaitu username dan password dalam bentuk string.

HASIL DAN PEMBAHASAN

Adapun flowchart untuk program yang diimplementasikan dapat dilihat pada gambar di bawah ini :



Gambar 1. Flowchart Program

1. Implementasi Solusi
   Implementasi dilakukan dengan menggunakan bahasa java dengan fungsi untuk mengenkripsi password dari akun yang telah dibuat. Fungsi untuk melakukan enkripsi password dapat dilihat pada gambar berikut:

```
public sha512() {
        frame = new
JFrame("Sign Up");
        frame.setDefaultClose
Operation(JFrame.EXIT_ON_CLOS
E);
        frame.setSize(300,
200);
        frame.setLayout(new
BorderLayout());

        // Username Panel
        JPanel usernamePanel
= new JPanel(new
FlowLayout());
        usernameLabel = new
JLabel("Username:");
        usernameTextField =
new JTextField(15);
        usernamePanel.add(use
rnameLabel);
        usernamePanel.add(use
rnameTextField);
 // Password Panel
        JPanel passwordPanel
= new JPanel(new
FlowLayout());
        passwordLabel = new
JLabel("Password:");
        passwordField = new
JPasswordField(15);
        passwordPanel.add(pas
swordLabel);
        passwordPanel.add(pas
swordField);
```

Gambar 2. Halaman sign up

Pada gambar di atas pembuatan halaman sign up berisi dengan sebuah panel username dan password yang nantinya akan diisi oleh pengguna. Tombol sign up dibuat untuk mengkonfirmasi bahwa proses pendaftaran akun akan dilakukan.

```
// Add action listener to the
sign up button
        signUpButton.addActionLis
tener(new ActionListener() {
            public void
actionPerformed(ActionEvent e) {
                String username =
usernameTextField.getText();
                char[] password =
passwordField.getPassword();

                // Encrypt
password using SHA-512
                String
encryptedPassword =
encryptSHA512(new
String(password));

                // Store the
account information in a file
                storeAccountInfo(
username, encryptedPassword);

                // Display
success message
                JOptionPane.showM
essageDialog(frame, "Account
created successfully!");

                // Clear the
input fields
                usernameTextField
.setText("");
                passwordField.set
Text("");
            }
        });

        frame.setVisible(true);
```

Gambar 3. Metode Action Listener

Setelah membuat halaman sign up selanjutnya pembuatan action listener yang berfungsi untuk mengambil nilai username dan password dari input fields, mengenkripsi password menggunakan SHA- 512, menyimpan informasi akun

TRIPLE A: Jurnal Pendidikan Teknologi Informasi. Vol. 2 (1), pp: 27 - 34

ke dalam file, menampilkan pesan keberhasilan, dan mengosongkan input fields untuk pendaftaran pengguna berikutnya

```
  private String
encryptSHA512(String input) {

        try {
            MessageDigest digest
= MessageDigest.getInstance("SHA-
512");
            byte[] hash =
digest.digest(input.getBytes(Stan
dardCharsets.UTF_8));

            // Convert byte array
to hexadecimal string
            StringBuilder
hexString = new StringBuilder();
            for (byte b : hash) {
                String hex =
Integer.toHexString(0xff & b);
                if (hex.length()
== 1) {
                    hexString.app
end('0');
                }
                hexString.append(
hex);
            }
            return
hexString.toString();
        } catch
(NoSuchAlgorithmException ex) {
            ex.printStackTrace();
        }
        return null;
    }
```

Gambar 4. Metode Enkripsi SHA-512

Ketika proses sign up dilakukan password akan masuk ke dalam proses enkripsi menggunakan algoritme hash SHA-512 dan mengembalikan hasil enkripsi sebagai string heksadesimal.

```
private void
storeAccountInfo(String username,
String password) {

        try (FileWriter writer =
new
FileWriter("C:\\Users\\Muhammad
Anum F\\Documents\\PNJ\\Sem 6
Java\\Anum\\src\\kripto\\accounts
.txt", true)) {
            writer.write(username
+ "," + password +
System.lineSeparator());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

Gambar 5. Metode Penyimpanan Data

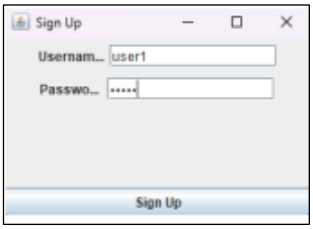
Proses terakhir adalah penyimpanan data akun ke dalam sebuah file dengan cara membuka file "accounts.txt" dan menulis informasi akun (username dan password) ke dalam file tersebut.

2. Pengujian

Pengujian dilakukan dengan dua tahap. Pengujian pertama adalah pengujian fungsionalitas program dan tahap kedua adalah pengujian waktu enkripsi.

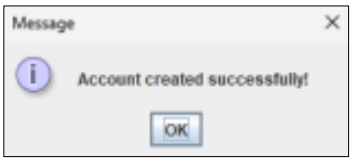
a. Pengujian Fungsionalitas

Tahap pengujian fungsionalitas dilakukan dengan cara melakukan skenario proses pembuatan akun dan melihat hasil data yang tersimpan



Gambar 6. Halam Sign Up

Pada tahap pertama pengguna akan diminta untuk memasukan user name dan password setelah mengisi, pengguna akan menekan tombol sign up.



Gambar 7. Pop Up

Pemberitahuan Setelah menekan tombol sign up, akan muncul pop up atau pemberitahuan bahwa akun telah berhasil dibuat.

```
anum,c7ad44cbad762a5da0a452f9e854fd
c1e0e7a52a38015f23f3eab1d80b931dd47
2634dfac71cd34ebc35d16ab7fb8a90c81f9
75113d6c7538dc69dd8de9077ec

LATIP,b14361404c078ffd549c03db443c3fe
de2f3e534d73f78f77301ed97d4a436a9fd9
db05ee8b325c0ad36438b43fec8510c204f
c1c1edb21d0941c00e9e2c1ce2

user1,c7ad44cbad762a5da0a452f9e854fd
c1e0e7a52a38015f23f3eab1d80b931dd47
2634dfac71cd34ebc35d16ab7fb8a90c81f9
75113d6c7538dc69dd8de9077ec
```

Gambar 8. Hasil Data Tersimpan

Data akun yang berhasil disimpan akan tersimpan pada sebuah file bernama account yang berformat username, password dan dengan password yang sudah terenkripsi.

b. Pengujian Waktu Enkripsi

Pengujian waktu autentikasi digunakan untuk mengetahui waktu yang digunakan saat melakukan autentikasi dengan menggunakan algoritme hash SHA 512. Prosedur pengujian dilakukan sebanyak 10 kali percobaan. Pada akhir percobaan, dihitung waktu rata-rata yang dibutuhkan dalam melakukan enkripsi. Pada percobaan ini, proses autentikasi berjalan secara sekuensial sehingga proses autentikasi dijalankan secara satu per satu. Berikut adalah hasil percobaan yang dilakukan:

Tabel 2. Percobaan Waktu Enkripsi

Percobaan ke-	Waktu
1	980 ms
2	801 ms
3	828 ms
4	1,062 ms
5	674 ms
6	831 ms
7	670 ms
8	1,186 ms
9	897 ms
10	976 ms
Rata-rata 888,5 ms	

Berdasarkan tabel di atas, didapatkan rata-rata waktu yang diperlukan untuk melakukan enkripsi menggunakan algoritme SHA-512 adalah 888,5 ms. Dalam hal ini waktu yang dibutuhkan untuk melakukan enkripsi sangat cepat dan efisien.

SIMPULAN

Algoritme SHA memiliki peranan yang vital dalam banyak aplikasi keamanan informasional. Salah satunya yang diimplementasikan dalam penelitian ini sebagai fungsi penyimpanan data padaprogram berbasis Bahasa Java. Implementasi algoritma hashing SHA-512 pada sistem halaman sign up menggunakan Java merupakan langkah yang efektif dalam meningkatkan keamanan data pengguna karena dapat memberikan tingkat keamanan yang tinggi dalam melindungi data pengguna. Dengan mengubah kata sandi pengguna menjadi hash value yang unik dan tidak dapat dikembalikan seperti semua, dengan ini dapat mencegah akses yang tidak sah ke akun pengguna.Hasil yang didapatkan dari implementasi ini cukup memuaskan karena sistem enkripsi berjalan dengan baik dan juga memiliki performa yang cepat dan efisien.

DAFTAR PUSTAKA

A. Hłobaż, "Statistical Analysis of Enhanced SDEx Encryption Method Based on SHA-512 Hash Function," 2020 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 2020, pp. 1-6, doi: 10.1109/ICCCN49398.2020.9209663.

Firdaus, Muhammad Risqi. 2021. Analisis Penggunaan Algoritma Bcrypt dengan Garam (Salt) untuk Pengamanan Password dari Peretasan

H. N. Bhonge, M. K. Ambat and B. R. Chandavarkar, "An Experimental Evaluation of SHA-512 for Different Modes of Operation," 2020 11th International Conference on

Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6.

Ipdal, Muhammad. 2021. Analisa Metode SHA-512 Untuk Tanda Tangan Digital Pada File Video .

Jai Verma, Md Shahrukh, Mukul Krishna, Ruchi Goel. "A CRITICAL REVIEW ON CRYPTOGRAPHY AND HASHING ALGORITHM SHA-512". International Research Journal of Modernization in Engineering Technology and Science. Volume:03/Issue:12/December-2021.

Johanes. 2020. Analisis Fungsi Hash pada Java DigestUtils

Mulya, Megah. 2009. Penggunaan Algoritma Sha-512 Untuk Menjamin Integritas Dan Keotentikan Pesan Pada Intranet

Munir, Rinaldi. 2020. Bahan Kuliah IF4020 Kriptografi: Fungsi Hash SHA

P. J. F. Bemida, A. M. Sison and R. P. Medina, "Modified SHA-512 Algorithm for Secured Password Hashing," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-9.

V. Sheketa, M. Pasieka, T. Serman, N. Pasieka, S. Chupakhina and L. Krul, "System Analysis and Example of Using SHA-512 Hash Functions to Protect Students' Personal Data on Educational Platforms," 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), LVIV, Ukraine, 2021, pp. 132-136, doi: 10.1109/CSIT52700.2021.9648699.