JURNAL INFORMATIKA DAN TEKONOLOGI KOMPUTER
Halaman Jurnal: https://journal.amikveteran.ac.id/index.php/jitek
Halaman UTAMA Jurnal : https://journal.amikveteran.ac.id/index.php

DOI : https://doi.org/10.55606/jitek.v3i2.1736

IMPLEMENTASI ALGORITMA SHA UNTUK ENKRIPSI TEKS BERBASIS MOBILE

Ananda Perdana [a], Dermawan Rizky Syahputra [b], Samuel Fernando Manurung [c], Taufiqul Hafizh [d], Dedek Indra Gunawan Hutasuhut [e*]

[a] Informatika, nandakeleng123@gmail.com, Universitas Potensi Utama
[b] Informatika, darmarizky2807@gmail.com, Universitas Potensi Utama
[c] Informatika, samuelfernando453@gmail.com, Universitas Potensi Utama
[d] Informatika, taufiqulhafizh30@gmail.com, Universitas Potensi Utama
[e] Informatika, dedek.indra@gmail.com, Universitas Potensi Utama
[*] Correspondence

ABSTRACT

In today's digital era, text messages have become one of the most common communication methods used by many people around the world. However, the use of text messages also poses security risks that need to be considered. Text messages can be targeted by hackers who are after personal information or important data that is sent through text messages. Mobile devices that are always connected to the internet can make text messages easily hacked or intercepted by unauthorized parties. Therefore, the use of encryption methods on text messages is very important to enhance the security and privacy of the information sent through text messages. This research aims to implement the Secure Hash Algorithm (SHA) algorithm on a mobile application for text encryption. The implementation of the SHA algorithm on the mobile application is done using the Java programming language and utilizing Android's built-in library to process data. The results of the study show that the mobile application using the implementation of the SHA algorithm can provide higher security for data containing messages sent over public networks

Keywords: *SHA Algorhytm. Java, Encryption*

Abstrak

Dalam era digital saat ini, pesan teks telah menjadi salah satu metode komunikasi yang paling umum digunakan oleh banyak orang di seluruh dunia. Namun, penggunaan pesan teks juga memiliki risiko keamanan yang perlu diperhatikan. Pesan teks bisa terkena serangan dari hacker yang mengintai informasi pribadi atau data penting yang dikirimkan melalui pesan teks. Perangkat mobile yang selalu terhubung ke jaringan internet dapat menyebabkan pesan teks mudah diretas atau disadap oleh pihak yang tidak berwenang. Oleh karena itu, penggunaan metode enkripsi pada pesan teks sangat penting untuk meningkatkan keamanan dan privasi informasi yang dikirimkan melalui pesan teks. Penelitian ini bertujuan untuk mengimplementasikan algoritma Secure Hash Algorithm (SHA) pada aplikasi mobile untuk melakukan enkripsi teks. Implementasi algoritma SHA pada aplikasi mobile dilakukan dengan menggunakan bahasa pemrograman Java dan memanfaatkan library bawaan Android untuk mengolah data. Hasil dari penelitian menunjukkan bahwa aplikasi mobile yang menggunakan implementasi algoritma SHA dapat memberikan keamanan yang lebih tinggi terhadap data berisi pesan yang dikirimkan melalui jaringan publik.

Kata Kunci: Algoritma SHA, Java, Enkripsi

1. PENDAHULUAN

Penggunaan perangkat mobile semakin meluas dalam kehidupan sehari-hari, termasuk dalam pengiriman pesan dan data sensitif. Oleh karena itu, keamanan data yang terkirim melalui perangkat mobile menjadi sangat penting. Salah satu cara untuk meningkatkan keamanan data adalah dengan menggunakan algoritma enkripsi yang kuat seperti algoritma SHA.

Received Juni 17, 2023; Revised Juni 27, 2023; Accepted Juli 15, 2023

SHA (Secure Hash Algorithm) adalah sebuah algoritma hash kriptografi yang digunakan untuk menghasilkan nilai hash unik dari suatu data. Nilai hash yang dihasilkan oleh SHA sangat sulit untuk direkonstruksi menjadi data asli yang digunakan sebagai inputnya. Oleh karena itu, SHA sering digunakan untuk keperluan enkripsi data.

Dalam jurnal ini, akan dibahas implementasi algoritma SHA yaitu SHA-1, SHA-224, dan SHA-256 untuk enkripsi teks berbasis mobile. Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi enkripsi teks berbasis mobile yang menggunakan algoritma SHA untuk meningkatkan keamanan data yang terkirim melalui perangkat mobile.

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan keamanan data yang terkirim melalui perangkat mobile. Selain itu, aplikasi enkripsi teks berbasis mobile yang dikembangkan juga dapat menjadi alternatif bagi pengguna yang membutuhkan keamanan data yang lebih baik dalam pengiriman pesan dan data sensitif melalui perangkat mobile.

2. METODOLOGI PENELITIAN

2.1 Algoritma SHA-1

Secure Hash Algorithm-1 (SHA-1) SHA dikembangkan oleh National Institute of Standards and Technology (NIST). SHA-1 adalah suatu teknik yang banyak digunakan dalam praktek enkripsi. Hash merupakan suatu metode yang secara langsung mengakses record-record dalam suatu tabel dengan melakukan perubahan aritmatik pada suatu input dari user yang biasanya merupakan bentuk string. Secure Hash Standard (SHS) menspesifikasikan SHA-1 untuk menghitung nilai hash dari sebuah pesan atau file. SHA-1 memiliki panjang pesan maksimal 264 bits dan memiliki keluaran sebesar 160 bits yang dinamakan Message digest.

2.2 Algoritma SHA-224

SHA-224 (Secure Hash Algorithm 224) adalah algoritma hash kriptografik yang mengonversi input data menjadi nilai hash 224-bit yang unik. SHA-224 mirip dengan SHA-256, namun dengan nilai awal yang berbeda dan menghasilkan nilai hash yang lebih pendek.

2.3 Algortrima SHA-256

SHA-256 (Secure Hash Algorithm 256) adalah algoritma hash kriptografik yang mengonversi input data menjadi nilai hash 256-bit yang unik. Algoritma ini dikembangkan oleh National Security Agency (NSA) dan diterbitkan oleh National Institute of Standards and Technology (NIST) pada tahun 2001.

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini penulis mengambil 1 sample nama untuk diuji menggunakan algoritma SHA1, 224, dan 256 yaitu “taufik”. sehingga didapatkan hasil dari pengujian sebagai berikut:

Tabel 1. Hasil Enkripsi dari setiap jenis SHA

	SHA-1	SHA-224	SHA-256
taufiqul	15c07f98ec8f5b37f 2c5002b674b27bc8 a398058	118e3164db54fc3d742a40638 14fcc0ffe659bbb45fdae2944d 6e50e	1a9f2b340c762de9431a896cb2a7 8e0ae0de480879ad4b9a97b642e6 58e735

Pada contoh di atas, "taufiqul" dienkripsi menggunakan algoritma SHA-1 dan menghasilkan nilai hash dalam format heksadesimal. Nilai hash yang dihasilkan memiliki panjang tetap 160-bit. Selanjutnya menggunakan algoritma SHA-224 dan menghasilkan nilai hash dalam format heksadesimal. Nilai hash yang dihasilkan memiliki panjang tetap 224-bit. Dan yang terakhir dienkripsi menggunakan algoritma SHA-256 dan menghasilkan nilai hash dalam format heksadesimal. Nilai hash yang dihasilkan memiliki panjang tetap 256-bit.

Berikut data dari perbedaan karakteristik varian algortima SHA:

Tabel 2. Karakteristik varian algoritma SHA

Algoritma	**Ukuran Pesan (bit)**	**Ukuran Blok (bit)**	**Ukuran Word (bit)**	**Ukuran *Message Digest* (bit)**
SHA 1	$<2^{64}$	512	32	160
SHA 224	$<2^{64}$	512	32	224
SHA 256	$<2^{64}$	512	32	256

SHA-1 menggunakan rangkaian fungsi logika f0,f1,...,f79 dalam proses perhitungannya. Setiap fungsi ft dimana 0 <; t < 79, mengambil tiga kata 32-bit (x, y, dan z) dan menghasilkan satu kata 32-bit. Fungsi ft(x, y, z) didefinisikan sebagai berikut untuk 0 ≤ t ≤ 19 atau 20 ≤ t ≤ 39,40 ≤ t ≤ 50 dan 60 ≤ t ≤ 79 [2]:

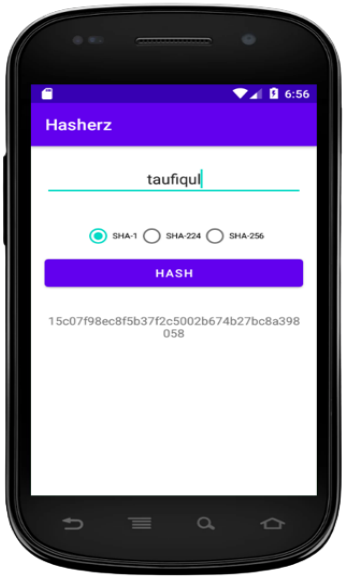
Ch(*x*, *y*, *z*) = (*x* Ù *y*) Å (Ø*x*^ *z*)
Parity(*x*, *y*, *z*) = *x* Å *y* Å *z*
Maj(*x*, *y*, *z*) = (*x* Ù *y*) Å (*x* Ù *z*) Å (*y* Ù *z*)
Parity(*x*, *y*, *z*) = *x* Å *y* Å *z*

Konstanta berikut (sebagai nilai heksadesimal) kemudian digunakan untuk menghitung hash pesan dengan SHA-1 [3]:

K(t) = 5A827999 (0 ≤ t ≤ 19)
K(t) = 6ED9EBA1 (20 ≤ t ≤ 39)
K(t) = 8F1BBCDC (40 ≤ t ≤ 59)
K(t) = CA62C1D6 (60 ≤ t ≤ 79)

Nilai *hash* awal (*initial hash value*) dari algoritma SHA-1 adalah sebagai berikut, sesuai yang telah ditetapkan pada [2]:

H0[0] = 67452301
H0[1] = efcdab89
H0[2] = 98badcfe
H0[3] = 10325476
H0[4] = c3d2e1f0



Gambar 1. Hasil enkripsi SHA1

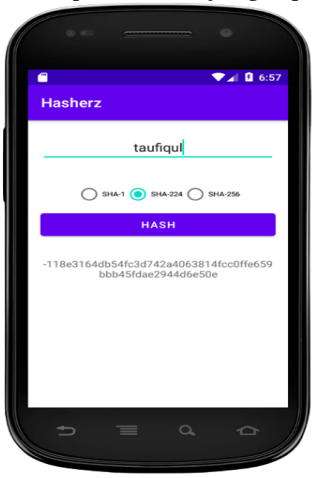
SHA-224 menggunakan enam fungsi logis di mana setiap fungsi beroperasi pada kata-kata 32-bit yang diwakili oleh x, y dan misalnya Hasil dari setiap fungsi adalah kata 32-bit baru

CH(x, y, z) = (x AND y) XOR ((NOT x) AND z)
MAJ(x, y, z) = (x AND y) XOR (x AND z) XOR (y AND z)
BSIG0(x) = ROTR^2(x) XOR ROTR^13(x) XOR ROTR^22(x)
BSIG1(x) = ROTR^6(x) XOR ROTR^11(x) XOR ROTR^25(x)
SSIG0(x) = ROTR^7(x) XOR ROTR^18(x) XOR SHR^3(x)
SSIG1(x) = ROTR^17(x) XOR ROTR^19(x) XOR SHR^10(x)

SHA-224 dan SHA-256 menggunakan urutan yang sama dengan enam puluh empat konstanta 32-bit Kata-kata, K0, K1, ..., K63. Kata-kata ini mewakili tiga puluh dua bit pertama dari pecahan tersebut akar pangkat tiga dari enam puluh empat pertama bilangan prima.

Untuk SHA-224, nilai hash awal, H (0), terdiri dari 32-bit kata-kata dalam hex, yaitu :
H(0)0 = C1059ED8
H(0)1 = 367CD507
H(0)2 = 3070DD17
H(0)3 = F70E5939
H(0)4 = FFC00B31
H(0)5 = 68581511
H(0)6 = 64F98FA7
H(0)7 = BEFA4FA4

Maka dengan mengikuti ketentuan di atas didapatkan hasil yang dapat dilihat pada Gambar 2:



Gambar 2. Hasil enkripsi SHA 224

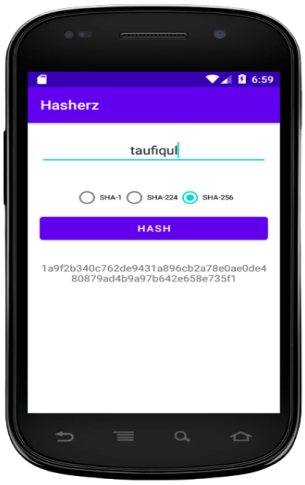
Dalam proses perhitungannya, SHA-256 menggunakan enam fungsi logika, masing-masing fungsi beroperasi pada tiga word 32-bit (x, y, dan z) dan keluarannya adalah satu word 32-bit. Berikut adalah karakteristik SHA-256[2]:

$Ch(x, y, z)$ = $(x$ Ù $y)$ Å $(Øx^{\wedge}$ $z)$
$Maj(x, y, z)$ = $(x$ Ù $y)$ Å $(x$ Ù $z)$ Å $(y$ Ù $z)$
$\sum_0^{256}(x)$ = $ROTR^2(x)$ Å $ROTR^{13}(x)$ Å $ROTR^{22}(x)$
$\sum_1^{256}(x)$ = $ROTR^6(x)$ Å $ROTR^{11}(x)$ Å $ROTR^{25}(x)$
$s^{256}(x)$ = $ROTR^7(x)$ Å $ROTR^{18}(x)$ Å $SHR^3(x)$
$s^{256}(x)$ = $ROTR^{17}(x)$ Å $ROTR^{19}(x)$ Å $SHR^{10}(x)$

Nilai *hash* awal pada algoritma SHA-256 adalah sebagai berikut:

H0[0] = 6a09e667
H0[1] = bb67ae85
H0[2] = 3c6ef372
H0[3] = a54ff53a
H0[4] = 510e527f
H0[5] = 9b05688c
H0[6] = 1f83d9ab
H0[7] = 5be0cd19

Maka dengan mengikuti ketentuan di atas didapatkan hasil yang dapat dilihat pada Gambar 3:



Gambar 3. Hasil enkripsi SHA 256

Maka dari penelitian yang dihasilkan ditarik kesimpulan yaitu:

a) Ketiga algoritma ini menggunakan teknik hash kriptografi yang sama, tetapi dengan ukuran output hash yang berbeda.
b) SHA-1 menghasilkan nilai hash dengan panjang tetap 160-bit, sedangkan SHA-224 menghasilkan nilai hash dengan panjang tetap 224-bit dan SHA-256 menghasilkan nilai hash dengan panjang tetap 256-bit.
c) Semakin panjang nilai hash, semakin sulit kemungkinan terjadinya tabrakan atau collision (situasi di mana dua data berbeda menghasilkan nilai hash yang sama).
d) Namun, SHA-1 dianggap sudah tidak aman untuk digunakan karena
e) SHA-1 dianggap sudah tidak aman untuk digunakan karena telah ditemukan kelemahan keamanan yang memungkinkan serangan collision yang lebih mudah dilakukan. Oleh karena itu, SHA-2 (termasuk SHA-224 dan SHA-256) disarankan sebagai pengganti SHA-1.
f) SHA-224 dan SHA-256 memiliki nilai hash yang lebih panjang daripada SHA-1, sehingga lebih sulit untuk dipalsukan atau diubah oleh pihak yang tidak berwenang.
g) Selain itu, SHA-256 juga dianggap lebih aman daripada SHA-224 karena memiliki panjang nilai hash yang lebih panjang, meskipun kedua algoritma tersebut masih dianggap aman dan dapat digunakan untuk kebanyakan kasus penggunaan.
h) Implementasi algoritma SHA-1, SHA-224, dan SHA-256 sangatlah mudah dan dapat dilakukan dengan menggunakan fungsi hash yang sudah disediakan oleh bahasa pemrograman. Namun, perlu diperhatikan bahwa nilai hash yang dihasilkan tidak dapat diubah atau didekripsi kembali menjadi data asli, sehingga harus diperlakukan dengan hati-hati dan tidak boleh dianggap sebagai bentuk enkripsi data.

Berikut implementasi source code algoritma SHA berbasis mobile android studio :

(Main Class)

```
package space.karaskiv.hasherz;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

import java.math.BigInteger;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

// TODO

    public void btnSHA(View v) {
        EditText etInput = findViewById(R.id.etInput);
        TextView tvOutput = findViewById(R.id.tvOutput);

        RadioGroup rg = findViewById(R.id.rg);
        int check = rg.getCheckedRadioButtonId();
        RadioButton rb = findViewById(check);


        byte[] inputData = etInput.getText().toString().getBytes();
        byte[] outputData = new byte[0];

        try {
            outputData = sha.encryptSHA(inputData, rb.getText().toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
        BigInteger shaData = new BigInteger(outputData);
        tvOutput.setText(shaData.toString(16));
    }
}
```

(Sha class)

```
package space.karaskiv.hasherz;

import java.security.MessageDigest;

public class sha {

    public static byte[] encryptSHA(byte[] data, String shaN) throws Exception {

        MessageDigest sha = MessageDigest.getInstance(shaN);
        sha.update(data);
```

```
    return sha.digest();
  }
    }
```

4. KESIMPULAN DAN SARAN

Penelitian ini membahas tentang implementasi algoritma SHA pada aplikasi mobile untuk menghasilkan hash value dari teks yang akan dienkripsi dan digunakan sebagai kunci enkripsi. SHA (Secure Hash Algorithm) adalah sebuah algoritma hash kriptografi yang digunakan untuk menghasilkan nilai hash unik dari suatu data. Nilai hash yang dihasilkan oleh SHA sangat sulit untuk direkonstruksi menjadi data asli yang digunakan sebagai inputnya. Oleh karena itu, SHA sering digunakan untuk keperluan enkripsi data.

Dari hasil penelitian yang telah dilakukan dengan menggunakan sempel text yaitu “Taufiqul” didapat hasil enkripsi menggunakan algoritma SHA sebagai berikut :

Tabel 3. Hasil Enkripsi dari setiap jenis SHA

	SHA-1	SHA-224	SHA-256
taufiqul	15c07f98ec8f5b37f2c5002b674b27bc8a398058	118e3164db54fc3d742a40638 14fcc0ffe659bbb45fdae2944d6e50e	1a9f2b340c762de9431a896cb2a7 8e0ae0de480879ad4b9a97b642e6 58e735

5. DAFTAR PUSTAKA

[1] Wibowo, S., Nilawati, E., & Suharnawi. 2014. Implementasi Enkripsi Dekripsi Algoritma Affine Cipher Berbasis Android. *Techno.COM* 13(4): p.215–221.

[2] Septiarini, A.,& Hamdani. 2011. Sistem Kriptografi Untuk Text Message Menggunakan Metode Affine. FMIPA Universitas Mulawarman.

[3] Juliadi, Prihandono, B., & Kusumastuti, N. 2013. Kriptografi Klasik Dengan Metode Modifikasi Affine Cipher Yang Diperkuat Dengan Vigenere Cipher. *Buletin Ilmiah Mat.Stat. dan Terapannya (Bimaster)* 2(2): p.87–92.

[4] Agung, H., & Budiman. 2015. Implementasi Affine Chiper Dan RC4 Pada Enkripsi File Tunggal. In Prosiding SNATIF ke-2 Tahun 2015, 243–250.

[5] Hossain, S.A., & Moniruzzaman, A. 2013. NoSQL Database : New Era of Databases for Big data Analytics- Classification , Characteristics and Comparison. International Journal of Database Theory and Application 6(4): p.1–13.

[6] Prasetyo, Ratno. 2016. Teknik Algoritma Kriptografi AES dan Fungsi Hash SHA-1 Berbasis Desktop.Universitas Budi Luhur

[7] Ibrahim, A. A. (2017) ‘Perancangan Pengamanan Data Menggunakan Algoritma AES (Advanced Encyption Standard)’, III(1), pp. 53–60.

[8] Android. *Budi Luhur Information Technology*, 13(1).Kurniawan, F., Kusyanti, A., & Nurwarsito, H. (2017). Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost. Jurnal Pengembangan *Teknologi Informasi dan Ilmu Komputer*. e-ISSN, 2548, 964X

[9] Prasetyo, T. F., & Hikmawan, A. (2016). Analisis Perbandingan Dan Implementasi Sistem Keamanan Data Menggunakan Metode Enkripsi RC4 SHA Dan MD5. *INFOTECH journal*, 2(1).

[10] Santoso, K. I. (2013). Dua Faktor Pengamanan Login Web Menggunakan Otentikasi One Time Password Dengan Hash SHA. *Semantik*, 3(1)

[11] Lusiana, Veronica, 2011, *Implementasi Kriptografi Menggunakan Algoritma AES-128.*

[12] Sutanto, Candra Alim, 2011, *Algoritma Fungsi Hash Baru dengan Menggabungkan MD5, SHA-1 dan Penyertaan Panjang Pesan Asli*, Bandung, Institut Teknologi Bandung.

[13] Widodo, Joko Tri Susilo, 2014, *Implementasi Algoritma Kriptografi AES 128 Bit Sebagai Pengaman SMS Pada Smartphone Berbasis Android*, Yogyakarta, Sekolah Tinggi Manajemen Informatika Dan Komputer AMIKOM Yogyakarta.

[14] W, I Putu Gede Darpana Putra, 2014, Perancangan dan Implementasi Aplikasi Kriptografi Pada Android dalam Pengamanan File Gambar dengan Menggunakan Algoritma SHA, Universitas Udayana.