GENERATE KUNCI SHA1, MD5, SHA-256 PADA JAVA ANDROID STUDIO

Viving Frendiana[1)], Fitri Elvira Ananda[2)]

[1,2]Teknik Elektro, Politeknik Negeri Jakarta, Jl. Prof. DR. G.A. Siwabessy, Kukusan, Beji, Depok, 16425
E-mail: viving.frendiana@elektro.pnj.ac.id

Abstract

Data security issues are a very serious problem in the use of information systems. Almost all the data we use is stored in a database that can be accessed online and can be used by any user. Data security can be done in several ways, for example limiting access rights and applying cryptographic methods. Hash algorithm, which is also called message digest algorithm is used to generate a special message digest for random messages. Secure Hash Algorithm (SHA) is a Hash function that is "irreversible" into the original message (one way) which will produce a checksum or fingerprint of the data. The Hash functions used for this research are SHA1, MD5, and SHA-256. On the other hand, accessing and retrieving information in this digital era is getting easier and faster. This development can also affect the way someone carries out the identity verification process to enter a system. The identity verification process that is widely used is using a fingerprint detection tool. It is hoped that Generating SHA1, MD5, SHA-256 Keys and Fingerprint Login Authentication in Java Android Studio Chipmunk can become a basic step in developing login security in the future.

Keywords: *SHA1, MD5, SHA-256, Android Studio, Java*

Abstrak

Masalah keamanan data merupakan masalah yang sangat serius dalam kegiatan penggunaan sistem informasi. Hampir semua data yang kita gunakan tersimpan dalam database yang dapat di akses secara online dan dapat dipergunakan oleh sembarang pengguna. Keamanan data dapat dilakukan dengan beberapa cara contohnya pembatasan hak akses serta menerapkan metode kriftografi. Algoritma hash, yang juga disebut algoritma intisari pesan digunakan untuk menghasilkan intisari pesan khusus untuk pesan acak. *Secure Hash Algorithm* (SHA) merupakan fungsi Hash yang bersifat "tidak dapat diubah kembali" menjadi pesan semula (satu arah) yang akan menghasilkan sebuah checksum atau *fingerprint* dari data tersebut. Fungsi Hash yang dipakai untuk penelitian ini adalah SHA1, MD5, dan SHA-256. Disisi lain, akses dan pengambilan informasi di era digital ini semakin mudah dan cepat. Perkembangan ini juga dapat mempengaruhi cara seseorang dalam melakukan proses verifikasi identitas untuk masuk ke dalam sebuah sistem. Proses verifikasi identitas yang sudah banyak digunakan adalah menggunakan alat deteksi sidik jari. Diharapkan dengan Generate Kunci SHA1, MD5, SHA-256 dan Otentifikasi Login Sidik Jari pada Java Android Studio Chipmunk dapat menjadi pijakan dasar dalam pengembangan keamanan login di masa mendatang.

Kata Kunci: *SHA1, MD5, SHA-256, Android Studio, Java*

PENDAHULUAN

Perkembangan teknologi informasi saat ini sangat pesat. Kebutuhan manusia akan akses informasi yang cepat menuntut kita untuk memanfaatkan teknologi yang ada saat

ini. Sejak adanya internet, informasi tidak lagi dibatasi. Fungsi internet sebagai gudang informasi adalah menyediakan informasi apapun, seperti informasi tentang apa saja yang ada di seluruh penjuru dunia, dan penggunaan teknologi informasi dan komunikasi sekarang menjadi cara transmisi informasi yang efektif dan efisien (Pamungkas et al., 2019)

Masalah keamanan data merupakan masalah yang sangat serius dalam kegiatan penggunaan sistem informasi di era Society 5.0. Hampir semua data yang kita gunakan tersimpan dalam database pada sebuah sistem informasi yang dapat di akses secara online dan dapat dipergunakan oleh sembarang pengguna (Customer, 2019).

Keamanan data dapat dilakukan dengan beberapa cara contohnya pembatasan hak akses, pengunaan nama field data yang hanya dipahami oleh pemilik aplikasi, serta menerapkan metode kriftografi pada aplikasi dengan tujuan field data yang disimpan menjadi lebih terjamin privasinya dan tidak dapat dimengerti oleh pihak luar mapun pihak dalam.

Kriptografi sendiri merupakan ilmu dan sekaligus seni untuk mengamankan data yang didalamnya terdapat algoritma tertentu yang bertujuan sebagai confusion atau pembingungan, dengan cara mengubah teks polos (plaintext) menjadi teks yang tidak bisa dibaca artinya secara langsung oleh manusia atau teks rahasia (ciphertext) (Idris et al, 2017). Kriptografi mempunyai proses enkripsi dimana dapat mengubah teks atau data (plaintext) menjadi teks rahasia (ciphertext), kemudian sebaliknya proses deskripsi yang dapat mengembalikan teks rahasia (ciphertext) menjadi teks atau data (plaintext)(Nasution et al, 2019).

Algoritma hash, yang juga disebut algoritma intisari pesan digunakan untuk menghasilkan intisari pesan khusus untuk pesan acak. Algoritma Hashing diklaim sebagai elemen penting dalam bidang kriptografi dan praktik keamanan. Hashing memiliki properti satu arah, dan karena properti inilah mereka diangap memiliki peran yang besar dalam memberikan integritas pesan dan penyimpanan kata sandi. Algoritma hash digunakan secara luas terutama dalam otentikasi login dan memverifikasi integritas pesan. Dalam hal ini, algoritma hash dapat membantu menjaga integritas pesan.

Secure Hash Algorithm (SHA) merupakan fungsi Hash yang bersifat "tidak dapat diubah kembali" menjadi pesan semula (satu arah) yang akan menghasilkan sebuah checksum atau fingerprint dari data tersebut. Umumnya dipergunakan untuk data

integration dan authentication. Kelebihan fungsi Hash yaitu menjaga integritas data, hemat dalam waktu pengiriman serta menormalkan panjang data yang beraneka ragam (Prasetyo et al, 2016). Fungsi Hash yang dipakai untuk penelitian ini adalah SHA1, MD5, dan SHA-256.

Disisi lain, akses dan pengambilan informasi di era digital ini semakin mudah dan cepat. Perkembangan ini juga dapat mempengaruhi cara seseorang dalam melakukan proses verifikasi identitas untuk masuk ke dalam sebuah sistem. Proses verifikasi identitas yang sudah banyak digunakan adalah menggunakan alat deteksi sidik jari (Rohman, 2020). Diharapkan dengan Generate Kunci SHA1, MD5, SHA-256 dan Otentifikasi Login Sidik Jari pada Java Android Studio Chipmunk dapat menjadi pijakan dasar dalam pengembangan keamanan login di masa mendatang.

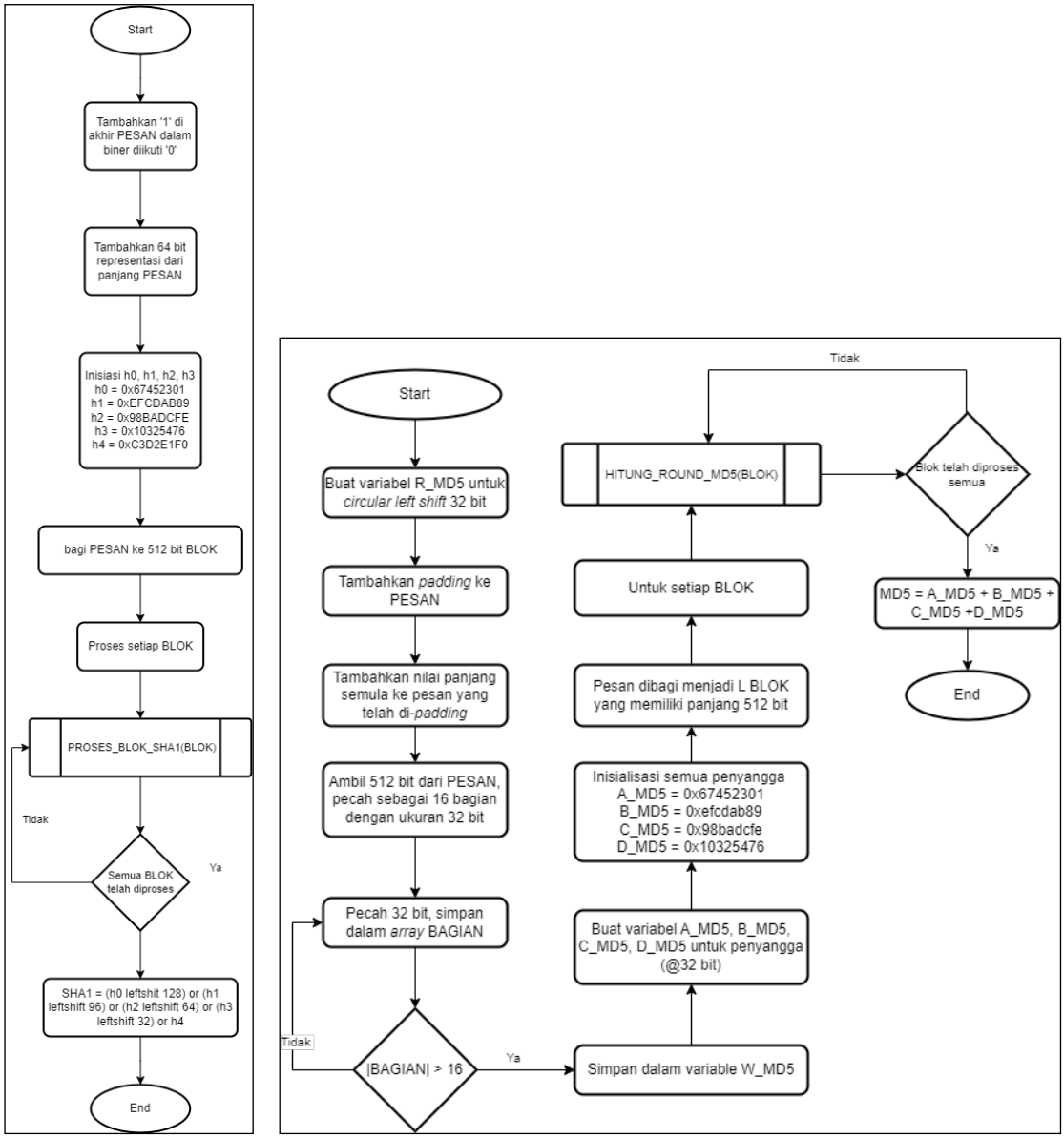
METODE PENELITIAN

SHA-1 atau *Secure Hash Algorithm* 1 adalah fungsi hash kriptografi yang mengambil input dan menghasilkan nilai hash 160-bit (20-byte). Nilai hash ini dikenal sebagai intisari pesan. Intisari pesan ini biasanya kemudian direnderi sebagai angka heksadesimal yang panjangnya 40 digit. Ini adalah Standar Pemrosesan Informasi Federal AS dan dirancang oleh Badan Keamanan Nasional Amerika Serikat. Untuk menghitung nilai hashing kriptografi di Java, Kelas MessageDigest digunakan, di bawah paket java.security. Kelas MessageDigest menyediakan fungsi hash kriptografi berikut untuk menemukan nilai hash dari sebuah teks.

Algoritma ini diinisialisasi dalam metode statis yang disebut getInstance(). Setelah memilih algoritma, nilai intisari pesan dihitung dan hasilnya dikembalikan sebagai array byte. Kelas BigInteger digunakan, untuk mengonversi larik byte yang dihasilkan menjadi representasi signumnya. Representasi ini kemudian diubah menjadi format heksadesimal untuk mendapatkan MessageDigest yang diharapkan. Contoh:

Masukan: halo dunia

Keluaran: 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed



(a) (b)

Gambar 1. (a) Flowchart SHA-1 (b) Flowchart MD5

Untuk menghitung nilai hashing kriptografi di Java, Kelas MessageDigest digunakan, di bawah paket java.security. MessageDigest menyediakan fungsi hash kriptografi berikut untuk menemukan nilai hash dari suatu teks. Algoritma ini diinisialisasi dalam metode statis yang disebut getInstance(). Setelah memilih algoritma, ia menghitung nilai intisari dan mengembalikan hasilnya dalam array byte. Representasi ini diubah menjadi format hex untuk mendapatkan MessageDigest. Contoh:

Masukan: halo dunia

Keluaran: 5eb63bbbe01eeed093cb22bb8f5acdc3

Gambar 2. Layout Desain Generate Key

Sebelum melakukan simulasi, penulis mendefinisikan konsepsi yaitu jumlah byte yang dienkripsi per satuan waktu. Pertama, penelitian ini membandingkan waktu berjalan; percobaan dilakukan pada Windows 10 dengan Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80GHz dan 12.00GB RAM dan sistem operasi 64-bit, x64-based processor. Lingkungan yang sedang berjalan digunakan untuk pengkodean java di Visual Studio Code versi 1.76.2. Setiap algoritma diuji string karakter sebanyak 10 kali.

HASIL DAN PEMBAHASAN

Tabel 1. Data Pengujian SHA-1

No	Karakter/String/Kata	Hasil Enskripsi
1	Hello World	0a4d55a8d778e5022fab701977c5d840bbc486d0
2	Selamat Datang	98c29ad22a66679a490196ec98d61128b423d2f2
3	Broadband	8288b9020e2a0b70ee6dddd0c7e16ba17620a095
4	Multimedia	40718650e06e46d91de6a770d91100a7a4565103
5	Jurusan	b228d94296f6f1c88aec86b92fe547d02fd69986
6	Teknik	3912d80ce4b3d10f99cd7c4b96d64694cadfae07
7	Elektro	2549e3ec98347f8aefaa0a9d4c88bf607ed1f0aa
8	Politeknik	c17b23143c0124d003a64d095dd8456341eb6824
9	Negeri	7fb0b51c2c2d6ac67786e5035e9705aa74b82b57
10	Jakarta	1d70ec60cc57dfc3ec4a1fe8fc862ad27ecb6723

SHA-1 (*Secure Hash Algorithm 1*) mengambil masukan dan menghasilkan nilai hash 160-bit (20 byte) sebagai intisari pesan dan diterjemahkan sebagai 40 digit heksadesimal. Hasil pengujian SHA-1 dilakukan sebanyak 10 kali ditunjukkan pada Tabel 1. Setiap karakter yang diinputkan akan dienkripsi menjadi karakter 40 digit heksadesimal.

Tabel 2. Data Pengujian MD-5

No	Karakter/String/Kata	Hasil Enskripsi
1	Hello World	b10a8db164e0754105b7a99be72e3fe5
2	Selamat Datang	7b1b93da6ee1880b79915f3c8f80fe29
3	Broadband	c6719d8b8f320640cdedcea10c458f37
4	Multimedia	2f56b4f336dc97edf739bf79523fb9a6
5	Jurusan	a5769ff0069e03afd285154f568e0bd7
6	Teknik	b91dd4eeb9de17bdd0e40a2ecaa3dde7
7	Elektro	cd6aaf81a5db8988342fdf1783b140ed
8	Politeknik	236ba91eb34f8c02a413e459c9f0f0b0
9	Negeri	0d8460764cbde4131fb6c94906b0ed8d
10	Jakarta	f3a693cf1392030d179eaa94d226f0ea

MD5 digunakan sebagai checksum untuk memverifikasi integritas data dan menghasilkan nilai hash 128-bit (16 byte) dan diterjemahkan menjadi 32 digit hexadecimal. Hasil pengujian MD5 dilakukan sebanyak 10 kali ditunjukkan pada Tabel 2. Pada Tabel 2 terlihat bahwa setiap karakter yang diinputkan berhasil dienkripsi menjadi karakter 32 digit heksadesimal.

Data pengujian kunci enkripsi SHA-256 ditunjukkan pada Tabel 3, dari hasil pengujian didapatkan bahwa algoritma kriptografi SHA-256 menghasilkan nilai hash 256-bit (32-byte) dengan panjang tetap. Tujuan dari algoritma SHA-256 adalah untuk membuat sidik jari digital unik dari suatu data.

Tabel 3. Data Pengujian SHA-256

No	Karakter/String/Kata	Hasil Enskripsi
1	Hello World	a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146e
2	Selamat Datang	8bcc613e79d8bb267c6845b473421675556de1103399c7e03db4f7510fe91ae9
3	Broadband	5f0a87a5759891bc80d0270a04dddeb57c329ee762b288c33761c50c5652c8a2
4	Multimedia	33f67dcf8b1d06ae76475137f6722c20e4fd5d653ee0986c1f516c077163a73d
5	Jurusan	4b80e3163a5eac5cf201f39bbed2813a6d8d1a8ca291564d3ac6f8b5fcc3032f
6	Teknik	5dedffe32144e8b2968f1c69a0b8f909e463a089a8cfac3ce72123120c56cb79
7	Elektro	7d00ee31494da6e1489e1572e09d4679c8f742bed6a31bd075c52f5e84e9db2a
8	Politeknik	9b30885154645e8153f5e555c8c092fadbfc1132c1285636e1c3e740fb74a7cc
9	Negeri	e39aa774cc7ec4edfd2c1d2d8019d21ada9fcdae0da462ab004f994615b81147
10	Jakarta	592e27e27927b8563b68af3bd4ee584077fe640d182dde33290375e9458d50cd

SIMPULAN

Kesimpulan dari penelitian Generate Kunci SHA-1, MD5, SHA-256 pada Java Android Studio berhasil dibuat dengan hasil berupa:

1. SHA-1 (*Secure Hash Algorithm 1*) menghasilkan nilai hash 160-bit (20 byte) sebagai intisari pesan dan diterjemahkan sebagai 40 digit heksadesimal.
2. MD5 menghasilkan nilai hash 128-bit (16 byte) dan diterjemahkan menjadi 32 digit hexadecimal.

3. Algoritma kriptografi SHA-256 menghasilkan nilai hash 256-bit (32-byte) dengan panjang tetap.

DAFTAR PUSTAKA

A. B. Nasution, "Implementasi Pengamanan Data Dengan Menggunakan Algoritma Caesar Cipher Dan Transposisi Cipher," J. Teknol. Inf., vol. 3, no. 1, p. 1, 2019, doi: 10.36294/jurti.v3i1.680

B. U. Customer, "Analisis Pengaruh Sistem Keamanan Informasi Perbankan pada Nasabah Pengguna Internet Banking," vol. 3, no. 1, pp. 1–9, 2019.

D Rachmawati, J T Tarigan and A B C Ginting, "A comparative study of Message Digest 5 (MD5) and SHA256 algorithm," Journal of Physics: Conference Series 2nd International Conference on Computing and Applied Informatics 2017

F. G. N. Larosa, J. F. Naibaho, and R. M. Tarigan, "Web Storage Berbasis Private Cloud Menggunakan Enkripsi Sha1," J. METHOMIKA, vol. 4, no. 1, pp. 56–59, 2020, [Online]. Available: http://www.methomika.net/index.php/jmika/article/view/142/81.

H. Rohman, U. Darussalam, N. D. Natashia, "Sistem Presensi Fingerprint Berbasis Smartphone Android," Jurnal Informatika Merdeka Pasuruan, Vol 5 No 1 Maret 2020.

Lase, H.,& Mufti. (2018). implementasi one time password (otp) mobile token dengan menggunakan metode algoritma MD5 dan SHA. Jurnal SKANIKA Vol.1 No.1, P.153

R. Pamungkas and S. Saifullah, "Evaluasi Kualitas Website Program Studi Sistem Informasi Universitas PGRI Madiun Menggunakan Webqual 4.0," INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf., vol. 3, no. 1, p. 22, Feb. 2019, doi: 10.29407/intensif.v3i1.12137.

R. Pamungkas, "OPTIMALISASI QUERY DALAM BASIS DATA MY SQL MENGGUNAKAN INDEX," Res. J. Comput. Inf. Syst. Technol. Manag., vol. 1, no. 1, pp. 27–31, 2018

R. Prasetyo and A. Suryana, "Aplikasi Pengamanan Data dengan Teknik Algoritma Kriptografi AES dan Fungsi Hash SHA-1 Berbasis Desktop," J. Sisfokom (Sistem Inf. dan Komputer), vol. 5, no. 2, p. 61, 2016, doi: 10.32736/sisfokom.v5i2.40.

Sihan Long, "A comparative Analysis of the Application of Hashing Encryption Algorithms for MD5, SHA-1, and SHA-512," Journal of Physics: Conference Series ICEMCE 2019

Y. Bin Idris, S. Adli Ismail, N. F. Mohd Azmi, A. Azmi, and A. Azizan, "Enhancement Data Integrity Checking Using Combination MD5 and SHA1 Algorithm in Hadoop Architecture," J. Comput. Sci. Comput. Math., vol. 7, no. 3, pp. 99–102, 2017, doi: 10.20967/jcscm.2017.03.007