Jurnal Mantik Penusa
Volume 4, No. 1, Juni 2020
e-ISSN 2580-9741
p-ISSN 2088-3943

PERANCANGAN APLIKASI HASHING FILE MENGGUNAKAN SHA – 224 BERBASIS ANDROID

***Hendra Pasaribu[1], Muhammad Fauzi Harahap [2]**
[1,2]Program Studi Teknik Informatika Fakultas Teknologi dan Ilmu Komputer, Universitas Prima Indonesia
Sekip, Simpang Seikambing, Medan, Indonesia, 20111
E-mail: *kevinbadboyz@gmail.com

ABSTRACT

Nowadays the development of technology in the field of computer science is growing rapidly. With the rapid development of technology in the field of computer science, information and data require a security and confidentiality because the information becomes a very valuable item. However, with the development of technology there are also many threats of crime against files that users have. One way to maintain the authenticity of information and data is to use hash cryptography. The hash function is an algorithm that converts text / messages into a series of random characters that have the same number of characters. The hash function is used to guarantee the authentication service of messages or files. Algorithms that use hash functions include MD5, SHA-1, SHA-2 (SHA - 224, SHA - 256, SHA - 384, SHA - 512) and others that have their advantages and disadvantages. In designing this application, the cryptographic algorithm used is the SHA-224 algorithm, because SHA-224 has a summary of variable sizes. The file hashing application using the Android-based SHA-224 algorithm, can help users to maintain the authenticity of files from users anywhere and anytime. Users can also compare the authenticity of user files with files that have been hashed.

Keywords: hash function, one-way hash, SHA-224, file hash, Android

ABSTRAK

Saat ini perkembangan teknologi di bidang ilmu komputer semakin pesat. Dengan pesatnya perkembangan teknologi di bidang ilmu komputer maka suatu informasi maupun data memerlukan suatu keamanan dan kerahasian karena informasi itu menjadi barang yang sangat berharga sekali. Namun, dengan berkembangnya teknologi juga banyak ancaman kejahatan terhadap file yang pengguna miliki. Salah satu cara untuk menjaga keaslian suatu informasi maupun data adalah dengan menggunakan kriptografi hash. Fungsi hash merupakan sebuah algoritma yang mengubah text/message menjadi sederetan karakter acak yang memiliki jumlah karakter yang sama. Fungsi hash digunakan untuk menjamin servis otentikasi dari pesan atau file. Algoritma yang menggunakan fungsi hash antara lain MD5, SHA-1, SHA-2 (SHA – 224, SHA – 256, SHA – 384, SHA – 512) dan lain sebagainya yang mempunyai kelebihan dan kekurangan masing – masing. Dalam perancangan aplikasi ini, algoritma kriptografi yang digunakan adalah algoritma SHA – 224, karena SHA-224 mempunyai ringkasan ukuran variabel. Aplikasi hashing file menggunakan algoritma SHA-224 berbasis android, dapat membantu para pengguna untuk menjaga keaslian file dari pengguna dimana saja dan kapan saja. Pengguna juga dapat membandingkan keaslian file pengguna dengan file yang telah di hash.

Kata kunci: fungsi hash, hash satu arah, SHA - 224, hash file, Android

1. PENDAHULUAN

Saat ini perkembengan teknologi di bidang ilmu komputer semakin pesat. Dengan pesatnya perkembangan teknologi di bidang ilmu komputer maka suatu informasi maupun data memerlukan suatu keamanan dan kerahasian karena informasi itu menjadi barang yang sangat berharga sekali. Namun, dengan berkembangnya teknologi juga banyak ancaman kejahatan terhadap file yang pengguna miliki. Salah satu cara untuk menjaga keaslian suatu informasi maupun data adalah dengan menggunakan kriptografi Hash. Fungsi Hash merupakan sebuah algoritma yang mengubah text/message menjadi sederetan karakter acak yang memiliki jumlah karakter yang sama. Fungsi Hash digunakan untuk menjamin servis otentikasi dari pesan atau file. Selain itu hash memiliki nama lain yang juga dikenal luas yaitu “One-Way Function”, yang artinya tujuan hash memang mengubah sebuah pesan yang dapat dibaca menjadi pesan acak sama

seperti enkripsi, namun hal mendasar yang menjadi perbedaan dari hash merupakan pesan yang telah di acak tadi tidak dapat diubah kembali menjadi pesan yang seharusnya, inilah mengapa hash disebut sebagai "One-Way Function".

Dalam penelitian ini, algoritma kriptografi yang digunakan adalah algoritma SHA-2. Karena algoritma SHA – 2 mempunyai berbagai pilihan jumlah bit yang digunakan dan bit yang digunakan untuk merancang aplikasi ini adalah SHA-224, karena SHA-224 mempunyai ringkasan ukuran variabel. Dengan adanya penelitian diharapkan dapat membantu para pengguna untuk menjaga keaslian file dari pengguna dimana saja dan kapan saja dan pengguna juga dapat membandingkan keaslian file pengguna dengan file yang telah di hash.

2. ISI PENELITIAN

2.1 SHA-2 (SHA-224)

SHA – 224 memiliki 224-bit satu arah fungsi hash, yang disebut SHA-224. The National Institute of Standards and Technology (NIST) mengumumkan FIPS 180-2 Perubahan Pemberitahuan pada tanggal 28 Februari 2004 yang menentukan SHA-224 satu-arah fungsi hash. fungsi hash satu arah juga dikenal sebagai mencerna pesan. SHA-224 didasarkan pada SHA-256, yang 256-bit satu arah fungsi hash sudah ditentukan oleh NIST. Perhitungan nilai hash SHA-224 adalah dua langkah. Pertama, SHA-256 Nilai hash dihitung, kecuali bahwa nilai awal yang berbeda adalah bekas. Kedua, menghasilkan nilai hash 256-bit dipotong untuk 224 bit. NIST mengembangkan pedoman manajemen kunci kriptografi, dan NIST baru-baru ini menerbitkan sebuah rancangan untuk komentar. Lima keamanan tingkat dibahas dalam bimbingan: 80, 112, 128, 192, dan 256 bit keamanan. fungsi hash satu arah yang tersedia untuk semua ini tingkat kecuali satu. SHA-224 mengisi kekosongan ini. SHA-224 adalah satu arah Fungsi hash yang menyediakan 112 bit keamanan, yang merupakan diterima kekuatan Triple-DES [3DES] umumnya. Ini membuat SHA-224 satu-arah fungsi hash spesifikasi tersedia untuk komunitas internet, dan menerbitkan objek pengidentifikasi untuk digunakan dalam protokol berbasis ASN.1.

Sejak SHA-224 didasarkan pada SHA-256, kira-kira jumlah yang sama usaha dikonsumsi untuk menghitung SHA-224 atau SHA-256 mencerna message digest nilai. Meskipun SHA-224 dan SHA-256 memiliki kira-kira setara kompleksitas komputasi, SHA-224 adalah pilihan yang tepat untuk fungsi hash satu arah yang menyediakan 112 bit keamanan. Penggunaan nilai awal yang berbeda memastikan bahwa pesan SHA-256 terpotong nilai digest tidak bisa salah untuk pesan SHA-224 mencerna nilai dihitung pada data yang sama. Beberapa lingkungan penggunaan yang sensitif terhadap setiap octet yang ditularkan. Dalam kasus ini, lebih kecil (oleh 4 oktet) pesan nilai digest disediakan oleh SHA-224 adalah penting.

Beberapa lingkungan penggunaan yang sensitif terhadap setiap octet yang ditularkan. Dalam kasus ini, lebih kecil (oleh 4 oktet) pesan nilai digest disediakan oleh SHA-224 adalah penting. Pengamatan ini menyebabkan bimbingan berikut:

1. Saat memilih suite algoritma kriptografi yang menawarkan semua 112 bit kekuatan keamanan, SHA-224 adalah pilihan yang tepat untuk fungsi hash satu arah.
2. Ketika terseness bukan kriteria seleksi, penggunaan SHA-256 adalah alternatif pilihan untuk SHA-224.

SHA-224 dapat digunakan untuk menghitung nilai hash satu arah pada pesan yang panjangnya kurang dari 2 ^ 64 bit. SHA-224 yang menggunakan SHA-256 [SHA2]. Untuk menghitung hash satu arah nilai, SHA-256 menggunakan jadwal pesan kata enam puluh empat 32-bit, delapan variabel 32-bit bekerja, dan menghasilkan nilai hash dari delapan 32-bit kata-kata. fungsi didefinisikan dengan cara yang sama persis seperti SHA-256, dengan berikut dua pengecualian: Pertama, untuk SHA-224, nilai hash awal dari delapan 32-bit variabel bekerja, secara kolektif disebut H, terdiri dari Berikut delapan kata 32-bit (dalam hex) :

H_0 = C1059ED8 H_4 = FFC00B31
H_1 = 367CD507 H_5 = 68581511
H_2 = 3070DD17 H_6 = 64F98FA7
H_3 = F70E5939 H_7 = BEFA4FA4

Kedua, SHA-224 hanya memanfaatkan tujuh kata 32-bit pertama dalam SHA-256 hasil, membuang kata-kata 32-bit yang tersisa dalam hasil SHA-256. Artinya, nilai akhir dari H digunakan sebagai berikut, di mana || menunjukkan rangkaian :

H_0 || H_1 || H_2 || H_3 || H_4 || H_5 || H_6

2.2 Uji Vektor

Bagian ini mencakup tiga vektor uji. vektor uji ini dapat digunakan untuk menguji implementasi dari SHA-224.

1. Uji Vektor 1
   Biarkan pesan untuk hash be 24-bit ASCII string "abc", yang adalah setara dengan string biner berikut: 01100001 01100010 01100011
   Nilai SHA-224 hash (dalam hex):
   23097D22 3405D822 8642A477 BDA255B3 2AADBCE4 BDA0B3F7 E36C9DA7
2. Uji Vektor 2
   Biarkan pesan untuk hash be 448-bit ASCII string yang "Abcdbcdecdefdefgefghfghighijhijkijkljklmkl mnlmnomnopnopq". SHA-224 nilai hash adalah (dalam hex):

75388b16 512776cc 5dba5da1 fd890150 b0c6455c b4f58b19 52522525

3. Uji Vektor 3
   Biarkan pesan untuk hash be bentuk kode-biner dari ASCII String yang terdiri dari 1.000.000 pengulangan karakter "a". SHA-224 nilai hash adalah (dalam hex):
   20794655 980c91d8 bbb4c1ea 97618a4b f03f4258 1948b2ee 4ee7ad67

2.3 Message Padding and Parsing

Misalkan pesan memiliki panjang L <2 ^ 64. Sebelum itu adalah input ke Fungsi hash, pesan yang empuk di sebelah kanan sebagai berikut:

1. "1" ditambahkan. Contoh: jika pesan asli "01010000", ini empuk untuk "010100001".
2. K "0" s yang ditambahkan di mana K adalah yang terkecil, non-negatif solusi untuk persamaan :
   $$L + 1 + K = 448 \ (mod\ 512)$$
3. Kemudian tambahkan blok 64-bit yang L dalam representasi biner. Setelah menambahkan blok ini, panjang pesan akan kelipatan 512 bit.
   Contoh: Misalkan pesan asli adalah bit string
   01100001 01100010 01100011 01100100 01100101
   Setelah langkah (a), ini memberikan :
   01100001 01100010 01100011 01100100 01100101 1
   Sejak L = 40, jumlah bit di atas adalah 41 dan K = 407 "0" s yang ditambahkan, membuat total sekarang 448. Hal ini memberikan berikut di hex:
   61626364 65800000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
   64-bit representasi L = 40 hex 00000000 00000028. Oleh karena itu pesan empuk akhir adalah hex berikut:
   61626364 65800000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000028

2.4 Fungsi dan Konstanta Digunakan

SHA-224 menggunakan enam fungsi logis, di mana masing masing fungsi beroperasi pada kata-kata 32-bit, yang direpresentasikan sebagai x, y, dan z. Itu Hasil setiap fungsi adalah 32-bit kata 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 dari enam puluh empat konstan 32-bit kata-kata, K0, K1, ..., K63. Kata-kata ini mewakili pertama tiga puluh dua bit dari bagian pecahan dari akar pangkat tiga dari pertama enam puluh empat bilangan prima. Dalam hex, kata-kata yang konstan adalah sebagai berikut (dari kiri ke kanan):

428A2F98	71374491	B5C0FBCF	E9B5DBA5
3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3
72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC
2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7
C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13
650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3
D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5
391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208
90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

2.5 Inisialisasi SHA - 224

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

2.6 Processing SHA - 224

SHA-224 melakukan pengolahan identik pada pesan blok dan hanya berbeda dalam cara H (0) diinisialisasi dan bagaimana mereka menghasilkan mereka hasil akhir. Mereka dapat digunakan untuk hash pesan, M, memiliki panjang L bit, di mana 0 <= L <2 ^ 64. Penggunaan Algoritma (1) jadwal pesan enam puluh empat kata 32-bit, (2) delapan variabel kerja 32 bit masing-masing, dan (3) nilai hash dari delapan kata 32-bit.

Kata-kata dari jadwal pesan diberi label W0, W1, ..., W63. Itu delapan variabel bekerja diberi label a, b, c, d, e, f, g, dan h. Itu kata-kata dari nilai hash diberi label H (i) 0, H (i) 1, ..., H (i) 7, yang akan terus nilai awal hash, H (0), digantikan oleh masing-masing berturut-turut antara nilai hash (setelah setiap blok pesan diproses), H (i), dan berakhir dengan nilai hash akhir, H (N), setelah semua blok N diproses. Mereka juga menggunakan dua kata sementara, T1 dan T2.

Pesan input empuk seperti dijelaskan di atas maka dipecah menjadi blok 512-bit, yang dianggap terdiri dari 16 32-bit kata M (i) 0, M (i) 1, ..., M (i) 15. Perhitungan berikut kemudian dilakukan untuk masing-masing blok pesan N. Semua Selain ini dilakukan modulo 2 ^ 32.

For i = 1 to N

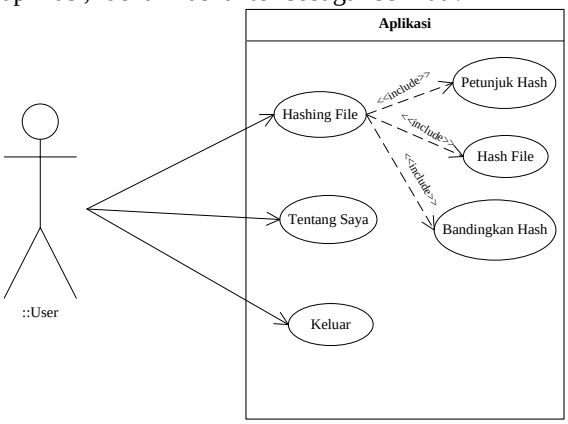
1. Siapkan jadwal pesan W :
   For t = 0 to 15
   Wt = M(i)t
   For t = 16 to 63
   Wt = SSIG1(W(t-2)) + W(t-7) + SSIG0(t-15) + W(t-16)
2. Menginisialisasi variabel bekerja :
   a = H(i-1)0
   b = H(i-1)1
   c = H(i-1)2
   d = H(i-1)3
   e = H(i-1)4
   f = H(i-1)5
   g = H(i-1)6
   h = H(i-1)7
3. Lakukan perhitungan hash utama :
   For t = 0 to 63
   T1 = h + BSIG1(e) + CH(e,f,g) + Kt + Wt
   T2 = BSIG0(a) + MAJ(a,b,c)
   h = g
   g = f
   f = e
   e = d + T1
   d = c
   c = b
   b = a
   a = T1 + T2
4. Menghitung antara nilai hash H (i) :
   H(i)0 = a + H(i-1)0
   H(i)1 = b + H(i-1)1
   H(i)2 = c + H(i-1)2
   H(i)3 = d + H(i-1)3
   H(i)4 = e + H(i-1)4
   H(i)5 = f + H(i-1)5
   H(i)6 = g + H(i-1)6
   H(i)7 = h + H(i-1)7

Setelah perhitungan di atas telah secara berurutan dilakukan untuk semua dari blok dalam pesan, hasil akhir dihitung. Untuk SHA-256, ini adalah gabungan dari semua H (N) 0, H (N) 1, melalui H (N) 7. Untuk SHA-224, ini adalah gabungan dari H (N) 0, H (N) 1, melalui H (N) 6.

3. HASIL DAN PEMBAHASAN

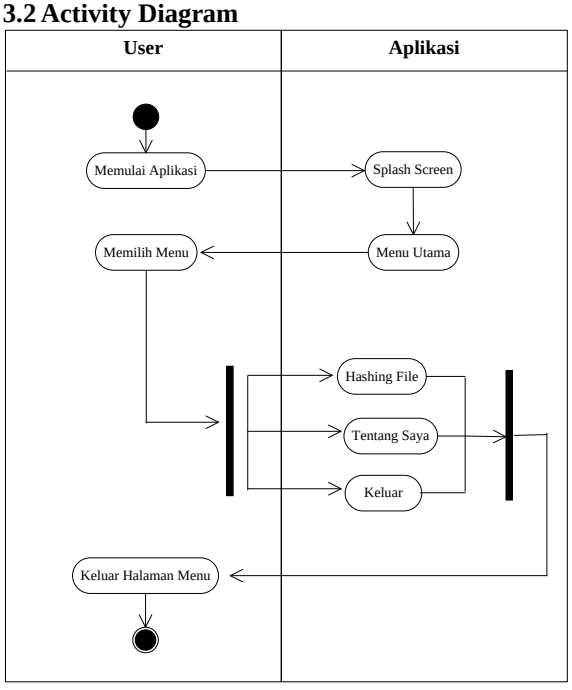
3.1 Use Case Diagram

Interaksi yang terjadi pada aplikasi hanya dilakukan oleh seorang user dimana user tersebut dapat memilih beberapa menu pilihan yang tersedia pada aplikasi, identifikasi aktor sebagai berikut :



GAMBAR I.
USE CASE DIAGRAM APLIKASI SUPER HASH

3.2 Activity Diagram



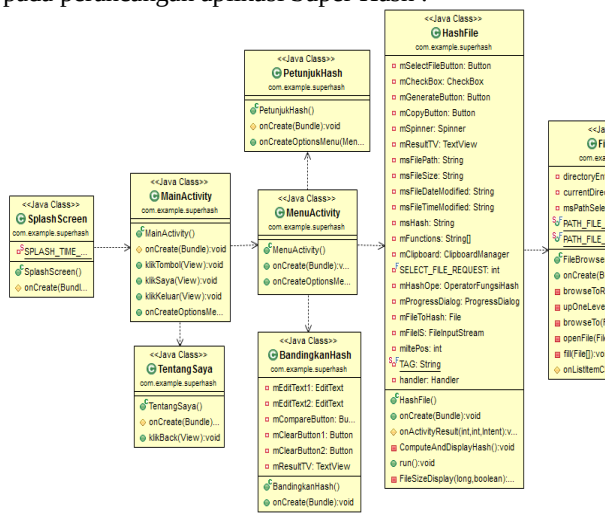
GAMBAR II.
ACTIVITY DIAGRAM MENU UTAMA

Adapun penjelasan dari gambar diatas adalah saat pengguna membuka aplikasi Super Hash, aplikasi akan menampilkan menu utama. Kemudian pengguna akan memilih menu - menu dari tombol

Hashing File, Tentang Saya atau Keluar untuk keluar dari aplikasi.

3.3 Class Diagram

Class Diagram menunjukkan interaksi antara kelas - kelas dalam sistem perancangan aplikasi Super Hash. Berikut adalah class diagram yang terdapat pada perancangan aplikasi Super Hash :



GAMBAR III.
CLASS DIAGRAM SUPER HASH

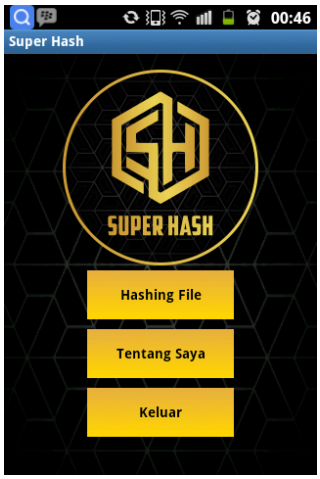
3.4 Kebutuhan Perangkat Lunak dan Perangkat Keras

Adapun kebutuhan aplikasi yang digunakan dalam perancangan Super Hash antara lain :

1. Perangkat keras/hardware yang digunakan dalam perancangan aplikasi Super Hash adalah sebagai berikut :
    1. Hardware yang digunakan pada komputer, antara lain :
        1. Processor minimal Intel Core i5 2.50 Ghz,
        2. RAM minimal 4 GB
        3. HardDisk minimal 500GB
        4. Monitor
        5. Keyboard dan Mouse
    2. Hardware dan software yang digunakan pada smartphone, antara lain :
        1. CPU Cortex A9 1 GHz
        2. RAM minimal 512 MB
        3. Display 4” WVGA (480 x 800)
        4. OS 2.3.3 Gingerbread
2. Perangkat lunak/software yang digunakan dalam perancangan aplikasi Super Hash adalah sebagai berikut :
    1. Sistem Operasi Windows 7
    2. Eclipse Juno
    3. Corel Draw X5
    4. JDK (Java Development Kit) Versi 1,7
3. Perangkat keras/hardware yang disarankan untuk menjalankan aplikasi Super Hash dengan spesifikasi smartphone yang disarankan adalah sebagai berikut:
    1. Processor minimal 800MHz
    2. RAM minimal 128MB
    3. Display minimal 5,3” WVGA (480 x 800)
4. Perangkat lunak/software yang disarankan untuk menjalankan aplikasi Super Hash yaitu minimal sistem operasi Android 2.3.3 (Gingerbread).

3.4 Hasil Implementasi

Berikut ini adalah tampilan utama dan tahapan langkah-langkah pengoperasian aplikasi Hashing File SHA-224.



GAMBAR IV.
MENU UTAMA SUPER HASH

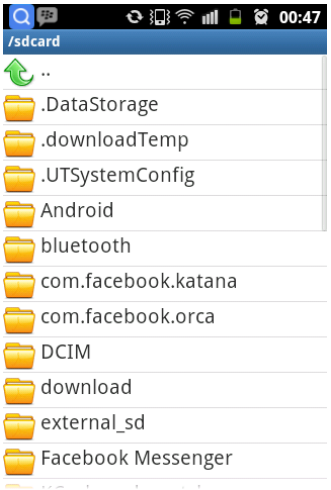
Tab ini menampilkan tentang tahap – tahap sebelum melakukan *hash file* :

GAMBAR V.
TAB PETUNJUK HASH

Menu akan menampilkan tempat untuk *Hashing File* dan tempat untuk menampilkan nilai *hash SHA – 224*.

GAMBAR VI.
MENU HASH FILE

Berikut ini adalah halaman *menu* untuk memilih *file* yang ingin di *hash*.



GAMBAR VII.
HALAMAN *MENU* PILIH *FILE*

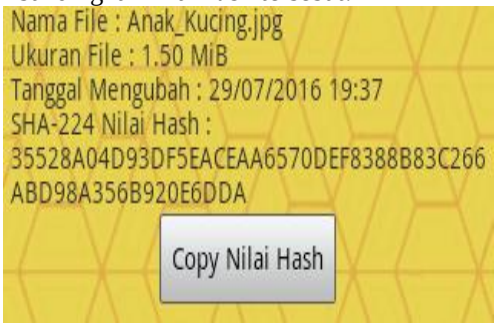
Berikut ini adalah halaman proses untuk *hash file* setelah *file* dipilih di halaman *menu* pilih *file*.

GAMBAR VIII.
FILE TELAH DIPILIH

Lalu, tahapan selanjutnya pilih tombol *Button* Hitung Hash untuk memulai menghitung nilai *hash* dari *file* yang telah dipilih.

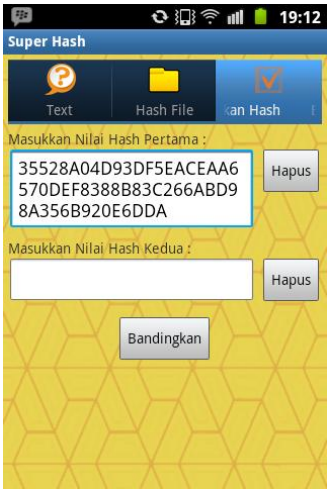
GAMBAR IX.
NILAI HASH FILE DALAM HURUF BESAR

Lalu, pilih *button* Copy Nilai Hash untuk meng*copy* nilai hash yang sudah ada, dan untuk membandingkan nilai *hash* tersebut.



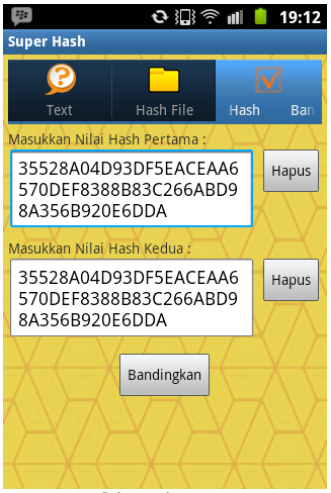
GAMBAR X.
COPY NILAI *HASH*

Pilih menu bandingkah *hash* untuk memulai membandingkan nilai *hash* yang telah di *copy* dan masukkan nilai *hash* ke dalam kolom.



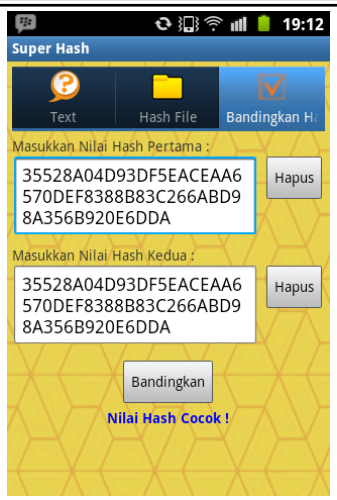
GAMBAR XI.
MENU BANDINGKAN HASH

Lalu, *copy* nilai *hash* ke dalam kolom Nilai Hash Kedua.



GAMBAR XII.
COPY NILAI *HASH* KE KOLOM KEDUA

Kemudian, pilih *button* Bandingkan untuk memulai membandingkan kedua nilai *hash* yang telah di *copy* dan bila nilai *hash* cocok akan tampil sebuah kalimat, begitu juga dengan nilai *hash* yang tidak cocok.



GAMBAR XIII.
NILAI HASH COCOK

Berikut ini adalah tabel tentang pengujian file berdasarkan durasi dalam melakukan hash file :

TABEL 1.
PENGUJIAN TERHADAP BEBERAPA FILE

No	Jenis File	File	Ukuran File	Durasi Melakukan Hash File
1.	mp4	Informasi_Bahaya_Mie.mp4	3,10 mb	1,81 detik
2.	mp4	Mentega.mp4	6,54 mb	2,68 detik
3.	mp4	Buat_Kue.mp4	7,80 mb	3,30 detik
4	jpg	Sate_Kacang.jpg	107, 74 kb	0,52 detik
5	jpg	Anak_Kucing.jpg	1,50 mb	1,22 detik
6	jpg	Kucing.jpg	1,83 mb	1,35 detik
7	pdf	Getting Started.pdf	806, 61kb	0,75 detik
8	pdf	Makalah.pdf	559, 40 kb	0,48 detik
9	pdf	Rumus Lengkap Fisika SMA.pdf	592,55 kb	0,52 detik
10	iso	Windows_XP_Professional.iso	617,81 mb	3, 50 menit
11	iso	WebForPC.Com_Windows.iso	2,53 gb	15, 56 menit
12	iso	7601.17514.10119-1850.iso	1,91 gb	11,55 menit

Dari pengujian durasi melakukan hash file adalah file yang memiliki ukuran lebih besar mempunyai durasi melakukan hash file lebih lama daripada file yang memiliki ukuran lebih kecil.

4. PENUTUP

4.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat diambil beberapa kesimpulan diantaranya:

1. Aplikasi hashing file SHA-224 ini telah berhasil melakukan hash file.
2. Cara hash suatu file dengan menggunakan algoritma SHA – 224 adalah dengan merubah file tersebut menjadi sederetan karakter acak yang memiliki jumlah karakter dengan panjang 56 bit
3. Aplikasi hashing file ini dapat menghindari proses pemalsuan data berupa file mp3, mp4, iso, txt, pdf, apk, png, jpg.

4.2 Saran

Sedangkan saran yang akan diusulkan untuk meningkatkan kualitas dari program tersebut dalam mencapai tingkat yang sempurna adalah sebagai berikut :

1. Aplikasi diharapkan dapat memeriksa apabila suatu file tersebut telah dilakukan hashing atau belum dan dapat mengunci suatu file yang telah dilakukan hashing.
2. Aplikasi diharapkan dapat menambahkan algoritma fungsi hash MD4, MD5, SHA – 1, SHA – 256, SHA – 384, SHA – 512 dan algoritma fungsi hash lainnya untuk melakukan hash file.

DAFTAR PUSTAKA

[1] Ricky Gilbert Fernando. “Penggunaan Fungsi Hash Dalam Kriptografi”. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2007.

[2] Abdurrisyad Fikri. “Analisis dan Perbandingan Algoritma Fungsi Hash SHA – 2 256 dan Keccak”. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2011.

[3] Somitra Kumar Sanadhya, Sarkar Palash. “New Collision Attacks Against Up To 24 – step SHA - 2”. Indian Statistical Institute, India, 2011.

[4] Christian Angga. “Analisis Cara Kerja Beragam Fungsi Hash Yang Ada”. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2011.

[5] Federal Information Processing Standards Publication. “Secure Hash Standard (SHS)”. Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, 2015.

[6] http://www.metode-algoritma.com/2013/06/fungsi-hash-pada-kriptografi.html diakses pada tanggal 02 Juni 2016

[7] http://www.metode-algoritma.com/2013/06/fungsi-hash-satu-arah-one-way-hash.html diakses pada tanggal 03 Juni 2016

[8] https://tools.ietf.org/html/rfc4634 diakses pada tanggal 12 Juni 2016

[9] https://tools.ietf.org/html/rfc3874 diakses pada tanggal 17 Juni 2016

[10] https://en.wikipedia.org/wiki/SHA-2 diakses pada tanggal 28 Juni 2016

[11] https://en.wikipedia.org/wiki/SHA-2 diakses pada tanggal 05 Juli 2016

[12] http://stackoverflow.com/ diakses pada tanggal 22 Juli 2016