Perancangan Sistemenkripsi Citra Digital dengan Algoritma Fungsi *Hash* Md5

[1]Komariyuli Anwariyah, [2]I Putu BagusDarmawanWicaksana,

Program Study Teknik Informatika, STMIK BumigoraMataram
Program Study Teknik Informatika, STMIK BumigoraMataram
Jalan Ismail MarzukiMataram, KodePos : 83121
[1]yuliaanwariyah@gmail.com, [2]thuagoes31@gmail.com,

Abstract

This application to find out how the algorithm Hash MD5 function to the input of bitmap image in the process of forming its hash value, perform the validation process of the image based on the hash value inserted in the picture, know the computing time to change the size of input image file, and build a desktop application. MD (Message Digest) version 5 is a widely used cryptographic hash function by processing the input file size indeterminate in length and generating 128-bit hash values. This research uses experimental method by designing and testing the design on computer. The design that is made is a desktop application to implement the Hash MD5 Function algorithm. The results showed that the pixel value of the bitmap image input is an input in the form of hexastream, the MD5 hash function algorithm regardless of the size of the input file will get the output of the hash value. The process of validating the authenticity of the image is done by comparing the hash value inserted in the image with the result hash value. View from the comparison of the original image with the image of the process, we get the difference of the image error which states that there is a change due to the insertion of the hash value. Computation time of MD5 hash value formation process to image file input size is not the same thing caused by change of input image file size significantly.

Keywords : *MD5 Hash function, hash value, message digest, bitmap*

I. PENDAHULUAN

Perkembangan teknologi saat ini sudah mempengaruhi segala lini kehidupan masyarakat. Semua orang bergantung pada teknolog iinformasi, berbagai proses termasuk pengiriman informasi baik informasi text ataupun gambar dilakukan melalui internet. Perkembangan teknologi ini tentunya memberikan kemudahan sekali resiko pagi para penggunanya.

Metode penyandian yang pertama kali dibuat masih menggunakan metode algoritma rahasia. Metode ini menumpukan keamanannya pada kerahasian algoritma yang digunakan. Namun metode ini tidak efisien saat digunakan untuk berkomunikasi dengan banyak orang. Oleh karena itu manusia tidak henti-hentinya berusaha menemukan cara-cara baru untuk menjamin keamanan dan kerahasiaan informasi yang dimiliki atau disebarluaskan. Salah satu perkembangan yang cukup signifikan dalam perkembangan algoritma kriptografi di zaman modern adalah perkembangan algoritma kriptografi dengan sistem Fungsi*hash*(Winarno, 2010).

Fungsi *hash* umumnya digunakan untuk keperluan autentikasi dan integritas data, fungsi *hash* merupakan fungsi yang bersifat satu arah dimana jika di masukkan data, maka fungsi *hash*a kanmenghasilkansebuah "*checksum*" atau "*fingerprint*" dari data tersebut dan tidak dapat diubah kembali menjadi pesan semula. Dilihat dari sisi matematik, fungsi hash memetakan satu set data kedalam sebuah set yang lebihkecil dan tetap ukurannya. Salah satu fungsi *hash* yang banyak digunakan dalam keamanan jaringan computer dan internet adalah *Message Digest* (MD) versi 5 atau di kenal dengan MD5.[1]

MD5 di desainoleh Ronald Rivest pada tahun 1991 untuk menggantikan fungsi *hash* sebelumnya, yakni MD4 yang telah di temukan

kecacatannya padatahun 1996 oleh Hans Dobbertin. MD5 merupakan fungsi *hash* kriptografik yang digunakan secara luas dengan mengolah ukuran file masukan yang tak tentu ukuran panjangnya dan menghasilkan nilai *hash* 128-bit. MD5 telah dimanfaatkan pada bermacam-macam aplikasi keamanan dan umumnya digunakan untuk melakukan pengujian integritas sebuahfile.[2]

Dalam skripsi ini, dilakukan eksperimen terhadap salah satu algoritma kriptografi fungsi *hash*, yaitu *Message Digest*versi 5 (MD5). Algoritma ini mendasarkan kekuatannya pada ketidak mungkinan untuk mendapatkan *plaintext* dari nilai *hash* yang di peroleh serta mendapatkan suatu nilai *hash* yang sama dari 2 file yang berbeda. Selain itu pada proses transformasi kriptografisnya fungsi*hash* MD5 tidakmembutuhkanstruktur data atau program yang komplekssehinggameningkatkankecepatan di dalam proses kerjanya. Selanjutnya dilakukan implementasi dalam bentuk perangkat lunak. Perangkat lunak yang dibangun yaitu aplikasi *desktopstandalone*fungsi*hash* MD5 serta di lakukanpengujianefisiensikecepatanterhadapukuran file.

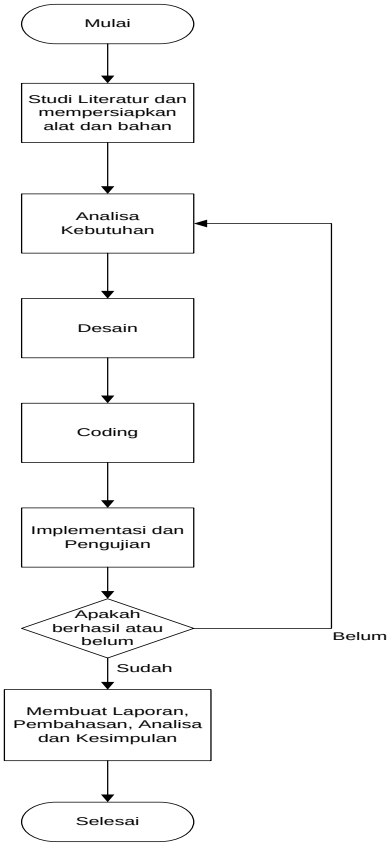
II. METODOLOGI

2.1. Metode Penelitian

Metode eksperimental yang merancang dan menguji rancangan. Rancangan yang dibuat adalah aplikasi *desktop standalone* untuk mengimplementasikan algoritma MD5.[3] Kemudian tahapan pada metodologi ini terdiri dari beberapa tahapan yaitu alat dan bahan, perancangan sistem, analisis kebutuhan, desain, implementasi, dan pengujian. Secara lebih detail, tahapan-tahapan metodologi tersebut dijelaskan seperti dibawah ini :

1. Perancangan Sistem
   Perancangan sistem dari aplikasi ini melandasi pembentukan algoritma fungsi hash MD5 dan melakukan pemrosessan validasi file gambar melalui hash value yang dihasilkan sebelumnya.
2. Analisis Kebutuhan
   Analisis kebutuhan meliputi elemen-elemen apa saja yang digunakan untuk membangun aplikasi *standalone* ini serta Alat dan Bahan yang digunakan sebagai rujukan dalam pembuatan aplikasi, dimana dalam prosedur proses penelitian ini saya melakukan kegitan pra penelitian dan pada saat penelitian berlangsung.
3. Desain
   Tahap *desain* ini lebih mengarah pada beberapa prosedur proses pembentukan nilai *hash* sampai dengan pengecekan *hash value* dengan file gambar.
4. Implementasi dan Pengujian Sistem
   Dalam tahap ini proses pengkodean dimaksudkan untuk merealisasikan desain dengan alur skenario yang telah dibuat dan langsung mengimplementasikan fasilitas atau fitur-fitur yang telah dibuat, serta tahap pengujian disini dimana mengetahui ketepatan pengimplementasian algoritma yang ditanamkan kedalam aplikasi.

Adapun tahapan penelitian ini dapat dilihat pada diagram alir pada gambar berikut ini :

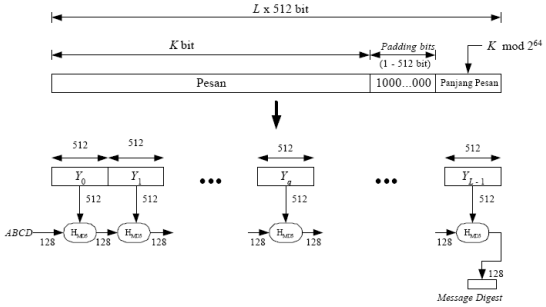


Gambar 2.1 Diagram Alir Penelitian

2.2 Metode Penyandian Pesan Dengan Kriptografi Algoritma MD5

Cara kerja kriptografi algoritma MD5 adalah menerima input berupa pesan dengan ukuran

sembarang dan menghasilkan *message diggest* yang memiliki panjang 128 bit. Berikut illustrasi gambar dari pembuatan *message diggest* pada kriptografi algoritma MD5 [4]:



Gambar 2.2 Pembuatan message digest dengan algoritma MD5

Memilik dari gambar diatas, secara garis besar pembuatan *message digest* ditempuh melalui empat langkah, yaitu :

1. Penambahan bit bit pengganjal

   Proses pertama yang dilakukan adalah menambahkan pesan dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512. Ini berarti setelah menambahkan bit-bit pengganjal, kini panjang pesan adalah 64 bit kurang dari kelipatan 512. Hal yang perlu diingat adalah angka 512 muncul karena algoritma MD5 memproses pesan dalam blok-blok yang berukuran 512.

   Apabila terdapat pesan dengan panjang 448 bit, maka pesan tersebut akan tetap ditambahkan dengan bit-bit pengganjal. Pesan akan ditambahkan dengan 512 bit menjadi 960 bit. Jadi panjang bit-bit pengganjal adalah antara 1 sampai 512. Lalu satu hal lagi yang perlu diperhatikan adalah bahwasanya bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

2. Penambahan nilai panjang pesan semula

   Kemudian proses berikutnya adalah pesan ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Apabila panjang pesan lebih besar dari 2^{64} maka yang diambil adalah panjangnya dalam modulo 2^{64}. dengan kata lain, jika pada awalnya panjang pesan sama dengan K bit, maka 64 bit yang ditambahkan menyatakan K modulo 2^{64}. sehingga setelah proses kedua ini selesai dilakukan maka panjang pesan sekarang adalah 512 bit.

3. Inisialisasi penyangga MD

   Pada algoritma MD5 dibutuhkan empat buah penyangga atau buffer, secara berurut keempat nama penyangga diberi nama A, B, C dan D. Masing-masing penyangga memiliki panjang 32 bit. Sehingga panjang total :

   A = 32 bit
   B = 32 bit
   C = 32 bit
   D = 32 bit +

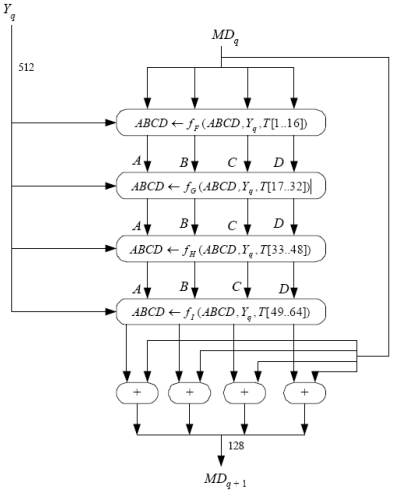
   Total = 128 bit

   Keempat penyangga ini menampung hasil antara dan hasil akhir. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi Hexadesimal) sebagai berikut :

   A = 01234567
   B = 89ABCDEF
   C = FEDCBA98
   D = 76543210

4. Pengolahan pesan dalam blok berukuran 512 bit

   Proses berikutnya adalah pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y_0 sampai Y_{L-1}). Setelah itu setiap blok 512 bit diproses bersama dengan penyangga MD yang menghasilkan keluaran 128 bit, dan ini disebut H_{MD5}. Berikut ini gambaran dari proses H_{MD5} :

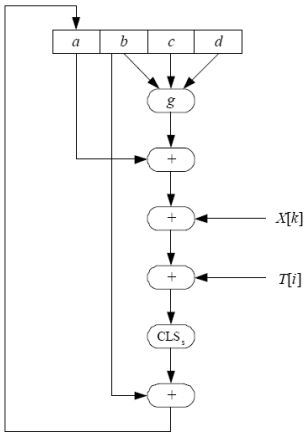


Gambar 2.3 Pengolahan blok 512 bit (Proses H_{MD5})

Dari gambar diatas dapat kita lihat bahwa proses H_{MD5} terdiri dari 4 buah putaran, dan masing-masing putaran melakukan opersi dasar MD5 sebanyak 16 kali. Dimana disetiap

operasi dasar memakai sebuah elemen T. Sehingga setiap putaran memakai 16 elemen tabel T.

Pada gambar 6, Y_q menyatakan blok 512 bit ke-q dari pesan yang telah ditambahkan dengan bit-bit pengganjal pada proses pertama dan tambahan 64 bit nilai panjang pesan semula pada proses kedua. MD_q adalah nilai *message digest* 128 bit dari proses H_{MD5} ke-q. Pada awal proses , MDq berisi nilai inisialisasi penyangga MD. Kemudian fungsi f_F,f_G, f_H, dan f_I pada gambar, masing-masing berisi 16 kali operasi dasar terhadap input, setiap operasi dasar menggunakn elemen tabel T. Berikut ini ilustrasi gambar operasi dasar MD5 :



Gambar 2.4 Operasi dasar MD5

1.1. Rancangan Flowchart Sistem

Sistem yang akan dibangun terdiri atas proses MD5, proses prosedur utama dimana dalam *flowchart* sistem ada 2 prosedur utama yakni proses pembentukan nilai hash dan proses validasi gambar bitmap (*.BMP) dan digambarkan dengan *flowchart* berikut :

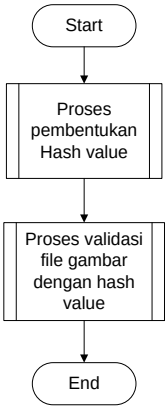
1.1.1. Proses algoritma MD5

Proses algoritma MD5 pada sistem ini yang akan di bangun langkah-langkahnya adalah sebagai berikut :

Gambar 2.5 Proses MD5

1.1.2. Proses Prosedur Utama

Proses aplikasi utama yang akan dibangun langkah-langkahnya adalah sebagai berikut :

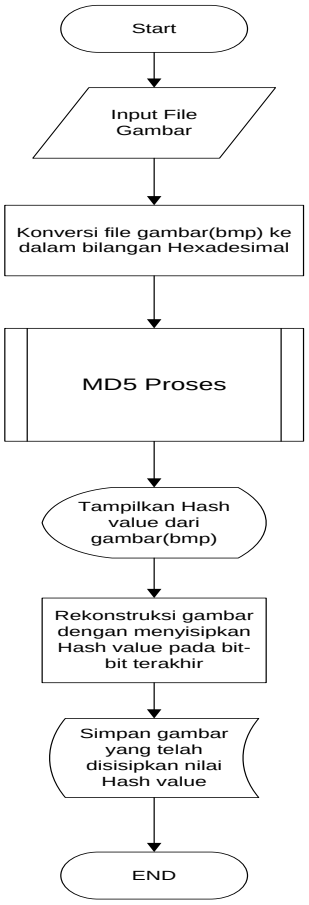


Gambar 2.6Prosedur Utama

Adapun proses diatas akan djelaskan secara detail pada langkah-langkah selanjutnya pada gambar 2.7 dangambar 2.8.

1.1.3. Proses Pembentukan *hash value*

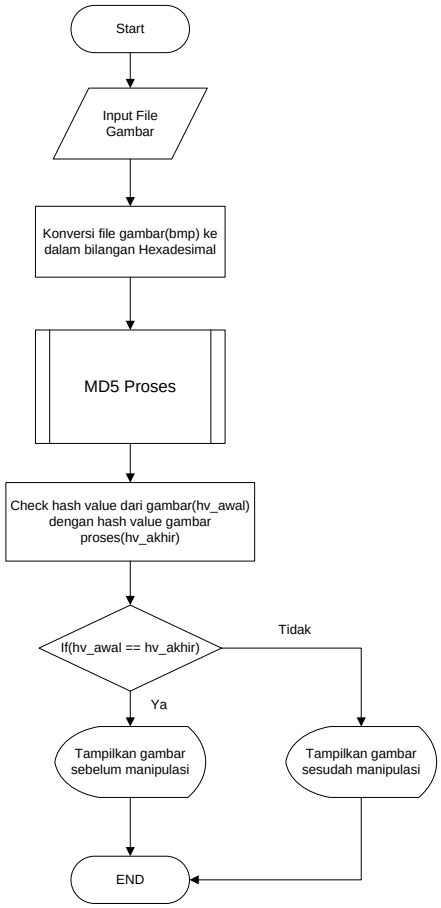
Proses pembentukan *hash value* pada sistem yang akan di bangun langkah-langkahnya adalah sebagai berikut :



Gambar 2.7Pembentukan *hash value*

1.1.3.1. Proses Validasi file gambar

Proses pengecekan validasi file gambar(*.bmp) dengan *hash value*nya merupakan salah satu fungsi pada sistem yang akan di bangun, langkah-langkahnya sebagai berikut :



Gambar 2.8Validasigambar(*.bmp) dengan *hash value*

II. HASIL DAN PEMBAHASAN

3.1 Proses KerjaSistem

Tahapan pengujian Aplikasi Enkripsi Keaslian Gambar Digital Menggunakan Algoritma Fungsi *Hash* MD5 ini menggunakan metode eksperimental, dimana pengujian aplikasi difokuskan pada inputan dan hasil output. Dimana pengujian aplikasi akan diambil 1 (satu) contoh gambar dengan ketentuan batasan masalah, agar dapat dilihat tingkat keakuratan hasil output dari penjabaran aplikasi ini. Pada proses pengujian gambar citra digital bitmap menampilkan proses pengambilan data gambar dengan ukuran 64 x 64 piksel yang akan diproses sampai hasil proses validasi. Aplikasi dimana terlihat semua proses dari gambar dimasukan, lalu proses enkripsi, validasi, deskripsi dan persentase error perbandingannya.

Gambar3.1

3.1.1 Informasi *Hexa Stream* dari Gambar Bitmap

Pada eksperimen di lakukan pembentukan nilai *hash* menggunakan gambar bitmap berukuran 64 x 64 pixel berikut :

Gambar 3.2 Gambar Asli

Untuk mendapatkan nilai perpixel dari gambar bitmap diatas di lakukan proses ekstrak dan di dapatkan informasi Bitmap info *header* sebagai berikut :

Tabel 3.1 Hasil informasi bitmap info *header*

Info *Header*	Nilai
iheader.size	12
iheader.width	64
iheader.height	64
iheader.bits	24
iheader.compression	0
iheader.imagesize	12344
iheader.ncolors	0
iheader.importantcolors	0

Data hasil informasi gambar bitmap pada eksperimen di peroleh sejumlah 12344 bilangan decimal dapat dilihat hasil informasi *hex value* dan nilai *hash* MD5 dalam bentuk hexa stream :

Tabel 3.2 Nilai Informasi Hexadesimal Value Gambar Asli

```
424D  3630  0000  0000  0000  3600
0000  2800  0000  4000  0000  4000
0000  0100  1800  0000  0000  0000
0000  120B  0000  120B  0000  0000
0000  0000  0000  171C  1A16  1A19
161B  1716  1B17  151A  1616  1B17
161B  1716  1B17  161A  1717  1B18
1919  1919  1919  1918  1819  1D19
191C  1919  1B19  1920  1B1A  2F26
1843  2D14  5831  174C  3D17  3631
1833 2219 1818 1918

1919  1918  1919  1819  1919  181C
1F16 2124 1839 ...
```

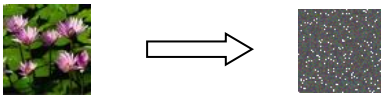
Berikut ini hasil dari nilai hash MD5 hasil proses gambar asli:

Tabel 3.3 Nilai Hash Md5 dari Gambar Asli

```
aZdosFo6QvsfoTlsdCvWP2DzSznqa178l
pjLetb21JCo+tDdAxygH2VG2K66ClWixr
j5gIeE7S0yoZGqV7RUl0YYfibvUqT1ADC
jUy+oT3R75glW2cR/QvkWfUZmIRXsZZdq
5VS3IoB8y/rlxHVwMmY2Gm21cwzoXlXsM
E/pgXLUhNsafHajakXvPw9xCVSrdmzNFw
O3Gi8ARAKTVSuA8jdmDYBVf+tuV/Ykikz
/XJCG2ZDxuI8f+49a3+ywXa5Y…
```

3.1.2 Penyisipan nilai hash MD5 padagambar bitmap

Proses penyisipan nilai *hash* MD5 pada gambar bitmap di lakukan pada bit akhir dari data gambar bitmap, nilai hash MD5 yang di sisipkan pada gambar berukuran 128 bit atau 16 pixel. Berikut gambar bitmap pada saat sebelum dan sesudah disisipkan nilai hash MD5 :



Gambar 3.3 Gambar Hasil Enkripsi

Tabel 3.4Hasilinformasi bitmap info *header*

Info Header	Nilai
iheader.size	9
iheader.width	64
iheader.height	64
iheader.bits	24
iheader.compression	0
iheader.imagesize	9291
iheader.ncolors	0
iheader.importantcolors	0

Data hasil informasi gambar bitmap enkripsi pada eksperimen di peroleh sejumlah 9291 bilangan desimal yang merepresentasikan titik pixel pada gambar dalam bentuk *hexa stream* :

Tabel3.5 Nilai Proses Tiap Putaran Hash Md5

8950 4E47 0D0A 1A0A 0000 000D
4948 4452 0000 0040 0000 0040
0806 0000 00AA 6971 DE00 0000
0467 414D 4100 00B1 8F0B FC61
0500 0000 0970 4859 7300 000E
C300 000E C301 C76F A864 0000
23E0 4944 4154 785E 559B 318E
EDBA 7245 9918 F04C 3C00 CFC0
0929 C081 488A 52F0 61C0 0949
3113 1CFE DC80 27E0 C178 241E
8287 A1BF 96FA 36F0 0DBC 83DB
AF5B 4722 AB76 EDDA 5545 85DC
B6B7 ACF2 9678 BDB5 A4B7 A4F1
3E73 7FAF 6DBD E7D1 DE2B 3F6F
5CF1 CDF7 7AE3 7EBF...

Berikut ini adalah nilai *hash*MD5 hasil proses dari Gambar :

Tabel 3.6 Nilai Hash Md5 hasil Proses

Lo0VDUFnUg2cVJaV9CRgZoAK2cqBy90S9
EICtO1oi/frNpHQVeQX3jKctJQR7UPL4r
4Bl4a57VbbXa6FdNrY/RSjocw+5SH3NBj
tEhedZciK0lF9xvQXaRh19EWSKcK60JZ3
NBBbOGpghYX7P81vMlEOnIc1utxWRSu/W
1AijkG6RCPBtCfFugV71FY216gQbj/abv
xl7W+Hd93bc3Z7RMotRCVZB8I01hgVNN2
FBPCWcXq9Ef8a16ef3RZf+ENWUURLFSQu
2+/mFcqeGWjy+oPMteY549E2PJzPUy1B/
x2+xG6cVVcdmhKqj1RrbY5...

Proses penyisipan nilai hash MD5 pada gambar asli didapatkan hasil gambar enkripsi yang tidak sama karena adanya penambahan karakter *hexa* dari algoritma fungsi *hash* MD5. Proses selanjutnya yang dilakukan adalah membandingkan gambar asli dengan gambar hasil deskripsi. Berikut gambar asli pada saat sebelum dan sesudah disisipkan nilai hash MD5 dan gambar hasil deskripsi :



Pada gambar ini memiliki nilai deskripsi dari algoritma fungsi *hash* MD5 berupa *hexa stream* berikut :

Tabel 3.7 Nilai Deskripsi Gambar kembali semula

424D 3630 0000 0000 0000 3600
0000 2800 0000 4000 0000 4000
0000 0100 1800 0000 0000 0000
0000 120B 0000 120B 0000 0000
0000 0000 0000 171C 1A16 1A19
161B 1716 1B17 151A 1616 1B17
161B 1716 1B17 161A 1717 1B18
1919 1919 1919 1918 1819 1D19
191C 1919 1B19 1920 1B1A 2F26
1843 2D14 5831 174C 3D17 3631
1833 2219 1818 1918 1919 1918
1919 1819 1919 181C 1F16 2124
1839 2D18 422F 1660 4515...

Setelah dilakukan proses validasi/perbandingan dari gambar asli dengan gambar enkripsi maka gambar hasil deskripsi akan lebih identik dengan gambar aslinya. Data hasil ujicoba tersebut dapat dilihat dari gambar asli dengan gambar hasil validasi nilai informasi *hex value* dan nilai deskripsi MD5 sama atau *valid*, namun pada gambar enkripsi memiliki perbedaan hasil putaran hash hasil MD5 dikarenakan gambar enkripsi telah dienkripsi atau disisipkan dengan nilai hash penambahan bit padding yang mengakibatkan gambar tersebut berubah.

3.2 Proses Perhitungan Persentase Selisih Error

Berdasarkan data gambar hasil pengujian aplikasi, dapat dicari persentase selisih error dari gambar asli dengan gambar hasil enkripsi (*output*) dengan menggunakan rumus sebagai berikut :

Loop : x(x1), y(y1) = (jumafter - jumbefore) + selisih

$$\text{Persen} = \frac{\text{jumafter} - \text{jumbefore}}{\text{jumbefore}} \times 100\ \%$$

Sumber : (Sulistyo, 2009)

Adapun hasil persen selisih *error* adalah sebagai berikut :

Loop : x(0), y(0) = (80 - 78) + 2 = 4
Loop : x(1), y(0) = (170 - 154) + 18 = 34
Loop : x(2), y(0) = (250 - 231) + 37 = 56
Loop : x(3), y(0) = (329 - 309) + 57 = 77
…….. sampai dengan looping
Loop : x(63), y(63) = (357084 - 283215) + 73919875 = 73993744

Setelah proses mencari x1, y1 sampai dengan 64x64 akan dimasukan dalam proses persentase untuk mencari selisih error dari kedua gambar :

$$\text{Persen} = \frac{357084 - 283215}{283215} \times 100\ \%$$

= 0.260823049626609 x 100 %
= 26.0823049626609 %

Berikut ini adalah table uji coba untuk 20 gambar menggunakan aplikasi yang penulis rancang dan hasil dari persentase selisih error dari gambar :

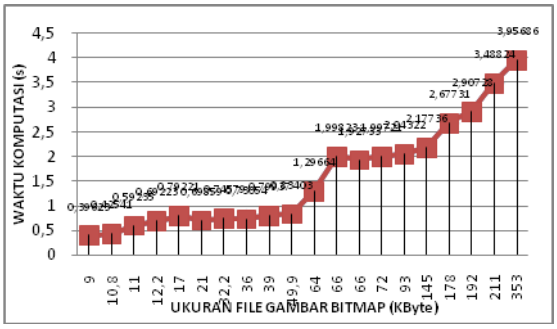
Table 3.8 Uji coba 20 gambar (*.bmp)

Ukuran Gambar (*.bmp)	Persentase Selisih Error
9	13.23%
10,8	26,08 %
11	39,93 %
12,2	14,80%
17	7,83%
21	25,78%
32,2	18,59%
36	32,76%
39	21,63%
49,9	24,43%
64	39,45%
66	27,65%
66	28.91%
72	21,86%
93	35,88%
145	21,70%
178	20,37%
192	22,49%
211	30,87%
353	33,29%

Dari file gambar diatas dapat disimpulkan rata-rata proses persentase selisih *error* dari pengujian 20 gambar adalah 507,53/20=25,3765% hal ini menunjukan kinerja dan efisiensi aplikasi ini sangat baik.

3.3 Pengujian Waktu Komputasi Proses Algoritma Fungsi Hash MD5 terhadap ukuran file masukan gambar BMP

Perhitungan waktu komputasi dan ukuran file gambar dilakukan dengan tujuan untuk mengetahui kinerja fungsi *hash* MD5 terhadap perubahan ukuran gambar bitmap dan berdasarkan jumlah bit per pixelnya.Dimana semakin besar ukuran gambar maka waktu yang diperlukan akan semakin besar namun peningkatan waktu enkripsi tidak signifikan paling tinggi waktu komputasi yang diperlukan 3,95686 detik dengan ukuran gambar 353kb. Rata-rata waktu proses untuk pengujian dari 20 gambar adalah 31,17643/20=1,5588215 ms. Hal ini menunjukan kinerja dan efisiensi fungsi *hash* MD5 sangat baik. Untuk lebih jelasnya dapat dilihat pada grafik berikut:



Gambar 3.5 Waktu Komputasi

IV. KESIMPULAN dan SARAN

Berdasarkan hasil penelitian yang didapatkan pada bab sebelumnya, maka dapat disimpulkan bahwa:

1. Algoritma fungsi *hash* MD5 merupakan fungsi *hash* satu arah yang menerima masukan hexadesimal dengan ukuran tidak tentu dan menghasilkan ringkasan pesan (nilai *hash*).
2. Algoritma fungsi *hash* MD5 memiliki sensitifitas yang baik karena perubahan 1 digit hexadesimal mengakibatkan perubahan pada keluaran nilai hash.
3. Algoritma fungsi *hash* MD5 sangat cocok di terapkan pada proses validasi gambar digital BMP karena perubahan 1 pixel pada gambar mengakibatkan perubahan hexadesimal pada masukan fungsi *hash*.
4. Dari gambar sebelum dan sesudah enkripsi ada terjadinya perubahan yang signifikan dikarenakan adanya penambahan bit-bit penganjal pada bit akhir gambar.
5. Algoritma fungsi hash MD5 sangat efisien di terapkan pada proses validasi gambar digital BMP karena semakin besar ukuran file gambar BMP, waktu komputasi yang di tunjukan oleh fungsi *hash* MD5 tidak berubah secara signifikan dan berpengaruh pada kecepatan aplikasi yang kita sedang pakai serta rata-rata waktu komputasi dari 20 gambar adalah 1,5588215 ms.
6. Berdasarkan penerapan algoritma yang dilakukan oleh penulis pada aplikasi ini yaitu proses pembentukan *hash value* MD5 dan validasi gambar dapat berjalan dengan baik serta berdasakan data gambar hasil pengujian aplikasi dapat dicari persentase selisih *error* rata-rata dari 20 gambar adalah 25,3765 %.

Berdasarkan pengembangan kedepanya yang lebih baik untuk Aplikasi Enkripsi Keaslian Gambar Digital Menggunakan Algoritma Fungsi Hash MD5 ini adalah sebagai berikut :

1. Dalam keterbatasan waktu dan *literatur* yang dimiliki oleh penulis dalam menyelesaikan aplikasi ini, maka diharapkan pada pengembang selanjutnya aplikasi ini dapat dikembangkan dengan inputan gambar yang memiliki format selain BMP.
2. Dengan kemajuan teknologi sekarang, aplikasi ini dapat dikembangkan menjadi aplikasi berbasis *Mobile* atau *Android.*

REFRENSI

[1] Aryasa, Komang., Paulus, Tommy, Yesaya., 2014. *Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java*. Citec Journal, Vol 1. No 1, 2354-5771.

[2] Dody, Andreanus., Suprihadi., Latuperissa, Rudy., 2011. *Sistem Pengamanan Data Menggunakan Metode MD5 dan Private Key pada Aplikasi Berbasis Client Server (Studi Kasus : KSP Buah Hati Bawen)*. Jurnal Teknologi Informasi-Aiti, Vol 8. No 2, 101 – 200.

[3] Danim, S. 2002. *Menjadi Peneliti Kualitatif.*Pustaka Setia, Bandung.

[4] Perdana, AdyaRahmat. 2008. *MetodePenyandianPesandanMenjagaIntegritas Data MenggunakanKriptografiAlgoritmafungsi Hash MD5*. Teknik Informatika, UniversitasSriwijaya.

Kadir, Abdul. 2003. *Pengenalan Sistem Informasi*.Andi Offset, Yogyakarta.

[5] Winarno, Edy. 2009. *Penyandian Data Dengan Kriptografi Password Based Encryption Menggunakan Message Digest 5 Dan Data Encryption Standart*. 145, 2085-3343.

[10] Wahana Komputer (Anonymous), 2011,. *Pemrograman Untuk Aplikasi Point of Sales.* Andi Offset, Yogyakarta.