Journal of Informatics, Electrical and Electronics Engineering
ISSN 2807-9507 (Media Online)
Vol 1, No 3, Maret 2022
Hal 110-116
https://djournals.com/jieee

Analisa Keaslian Tanda Tangan Dengan Menerapkan Algoritma Ripemd160

Jefri Sianipar

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: jefri 33245@gmail.com

Abstrak Tanda tangan merupakan salah satu ciri dari setiap orang. Tanda tangan banyak digunakan sebagai syarat untuk mengesahkan dokumen-dokumen legal. Hal itu menjadi bermasalah jika suatu transaksi bermasalah atau gagal karena adanya pemalsuan tanda tangan, tentu saja hal tersebut sangat merugikan, sehingga sangat penting untuk melakukan verifikasi tanda tangan. Salah satu cara untuk dapat mengidentifikasi kecocokan tanda tangan yaitu dengan memanfaatkan teknik pengolahan citra. Untuk mendapatkan identifikasi kecocokan tanda tangan tersebut digunakan metode RIPEMD160 yang mengutamakan pengolahan nilai Hash yang dimiliki oleh sebuah citra digital untuk memberikan ciri khas khusus bagi sebuah data citra digital, dan sekaligus sebagai sebuah data yang akan digunakan sebagai verifikator untuk keaslian dari data citra digital dengan mengolah citra tanda tangan dan memberikan informasi yang hanya dimiliki oleh tanda tangan yang asli. Hasil dari proses ini akan menyatakan cocok atau tidak cocok suatu tanda tangan dan dapat membedakan dirinya dengan tanda tangan yang sudah memiliki verifikasi asli dan bukan.

Kata Kunci: Otentikasi; Verifikasi; Tanda Tangan; Ripemd160

Abstract Signature is one of the characteristics of every person. Signatures are widely used as a condition for ratifying legal documents. This becomes problematic if a transaction is problematic or fails due to signature forgery, of course this is very detrimental, so it is very important to verify the signature. One way to be able to identify a match signature is by utilizing image processing techniques. To get the identification of the signature match, the RIPEMD160 method is used which prioritizes processing the Hash value of a digital image to provide special characteristics for a digital image data, and at the same time as a data that will be used as a verifier for the authenticity of digital image data by processing the signature image and provides information that only the original signature would have. The result of this process will indicate that a signature matches or does not match and can distinguish itself from a signature that has an original verification and is not.

Keywords: Authentication, Verification, Signature, Ripemd160

1. PENDAHULUAN

Pada abad ke 21 pengesahan dan verifikasi dokumen tidak lagi hanya menggunakan stempel sederhana atau cap biasa. Seperti contohnya tanda tangan, tanda tangan tidak lagi hanya berbentuk manual namun juga dalam bentuk digital. Penggunaannya juga mulai merambah pada dokumen dokumen yang krusial dan memiliki nilai ekonomi tinggi. Ini artinya penggunaan tanda tangan dalam bentuk citra digital sudah diakui oleh dunia. Tanda tangan berlaku sebagai segel. Tanda tangan sering digunakan untuk banyak hal yang sebagian besar bersifat legal hukum atau berkekuatan hukum dan memiliki nilai atau value dan juga dapat memberikan nilai pada berkas-berkas tertentu Tanda tangan dapat diduplikasi seperti aslinya tanpa mengurangi kualitas Tanda tangan asilnya.

Hal ini sulit dilakukan dalam teknologi analog, dimana kualitas tanda tangan asli lebih baik dari duplikatnya, sehingga seseorang dengan mudah dapat memverifikasi keaslian sebuah dokumen. Tetepi e-dokumen memiliki kelemahan yaitu sangat mudah untuk diduplikasikan sehingga tidak diketahui lagi Tanda tanganmana yang asli. E-dokumen sebagai alat bukti dikhawatirkan dapat dipalsukan dan nantinya akan muncul masalah tentang keaslian dokumen elektronik tersebut.Hal ini menyebabkan keaslian (autentifikasi) tanda tangan menjadi sangat penting untuk dilindungi dari orang-orang yang tidak bertanggung jawab yang akan memanipulasi tanda tangan untuk kepentingannya yang dapat merugikan orang lain.

Untuk memastikan sebuah dokumen asli atau tidak, terdapat beberapa metode dengan konsep otentifikasi, serta dapat mendeteksi perubahan dokumen dari hasil manipulasi salah satunya adalah algoritma RIPEMD160. Algoritma RIPEMD160 merupakan variasi dari algoritma MD4 dan peningkatan dari algoritma RIPEMD128. Algoritma RIPEMD160 juga merupakan suatu fungsi hash kriptografi yang cepat, dan dirancang untuk implementasi pada perangkat lunak dengan arsitektur 32- bit RIPEMD160 menghasilkan 160 bit dimana data tanda tangan yang sudah menjadi citra digital akan diproses untuk membangkitkan nilai Hash unik yang hanya akan dimiliki oleh dokumen tersebut.

Nilai hashyang ada kemudian akan disimpan atau didistribusikan melalui jalur yang aman dan akan digunakan seperti kode pin untuk melakukan verifikasi terhadap tanda tangan yang sudah ada, akan digunakan, atau yang sedang di proses didalam sebuah dokumen. Nilai hash akan mengalami perubahan yang amat signifikan apabila tanda tangan yang sudah digitalisasikan mengalami perubahan sekecil apapun, bahkan apabila perubahan yang dilakukan hanya satu piksel dari total citra akan memberikan nilai hash yang berbeda sehingga pemalsuan tanda tangan dapat dihindari dengan cukup efisien.

Salah satu masalah keamanan tersebut adalah pencurian dan pemalsuan data. Tanda tangan yang dikirim lewat internet dapat diambil dan diubah oleh orang yang tidak bertanggung jawab. Salah satu cara untuk mencegahnya adalah dengan membuat suatu tanda khusus yang memastikan bahwa Tanda tangan tersebut adalah Tanda tangan yang benar. Tanda tangan merupakan salah satu bukti untuk membenarkan atau mengesahkan suatu dokumen, dikarenakan keaslian

dokumen yang memiliki tanda tangan menjadi aset yang sangat berharga baik dari suatu organisasi, perusahaan, pemerintah maupun pribadi [1].

Pada penelitian sebelumnya yang dilakukan oleh Ferry Mulia yang dipublikasikan pada Program Studi Teknik Informatika, Institut Teknologi Bandung dengan judul "Studi, Analisis, Dan Implementasi Fungsi Hash Ripemd-160" dapat diambil kesimpulan bahwa algoritma RIPEMD-160 melakukan hashing terhadap pedan dan menghasilkan 160 bit message digest. Walaupun performansi algoritma ini kurang jika dibandingkan dengan algoritma lain, setidaknya fungsi ini masih tahan terhadap serangan-serangan yang biasanya dilakukan terhadap fungsi hash lainnya

2.1 Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan, namun pada pengertian modern kriptografi berarti ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentifikasi entitas [5]. Kriptografi juga dapat diartikan sebagai ilmu yang mempelajari tentang teknik matematika yang berhubungan dengan aspek keamanan informasi seperti tingkat keyakinan, integritas data, autentifikasi entitas dan autentifikasi keaslian data [6].

Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih dari sekadar *privacy*, tetapi juga untuk tujuan *data integrity*, *authentication*, dan *non-repudiation*.

2.2 Fungsi Hash

Fungsi *hash*(*hash function*) merupakan salah satu teknik kriptografi untuk menghitung nilai unik dari sebuah data. Fungsi *hash* dapat diibaratkan sebagai sidik jari elektronik berguna untuk menentukan orisinalitas sebuah dokumen elektronik. Dua dokumen elektronik yang berbeda akan memiliki nilai *hash*yang berbeda, itulah sebabnya apabila sebuah dokumen telah mengalami perubahan, maka nilai *hash* juga akan berubah. Sebuah dokumen dengan panjang berapapun akan menghasilakn nilai *hash* dengan panjang tertentu sesuai dengan spesifikasi fungsi *hash* yang digunakan [3].

2.3 Ripemd160

RIPEMD-160 adalah suatu fungsi hash kriptografi yang dirancang untuk implementasi pada perangkat lunak dengan arsitektur 32 bit. Desain utama dari fungsi hash ini ada dua proses komputasional yang berbeda dan saling independen, dimana hasil dari kedua proses ini akan digabungkan pada akhir perhitungan dengan sebuah fungsi kompresi. Sesuai dengan namanya RIPEMD-160 menghasilkan 160 bit. Hal ini dimaksudkan untuk menyediakan level sekuritas yang lebih tinggi untuk 10 tahun yang akan datang. Sama halnya dengan varian MD4 lainnya, RIPEMD-160 beroperasi pada prosesor 32-bit. Dibawah ini adalah deskripsi langkah-langkah proses perhitungan nilai *hash* pada algoritma RIPEMD-160 [2]:

1. Definisi fungsi, permutasi dan inisialisasi *buffer*.
   Sebelum proses perhitungan, terlebih dahuIu didefinikan fungsi-fungsi dan permutasi yang akan digunakan pada proses kompresi. Definisi kelima fungsi tersebut
   adalah : Kemudian disiapkan juga suatu vektor awal, yaitu *buffer* (H0, H1, H2, H3, H4), yang masing-masing nilainya dalam notasi heksadesimal adalah:
   H0 : 67452301x;
   H1 : EFCDAB89x
   H2 : 98BADCFEx;
   H3 : 10325476x;
   H4 : C3D2EIFOx
2. Proses Kompresi
   Bagian yang merupakan inti dari proses algoritma RIPEMD-16() ini, terdiri dari dua buah rantai fungsi, masing-masing terdiri dari 5 *round*, dimana setiap *round*-nya terdiri dari 16 operasi, yang dirangkai secara paralel. Langkah-langkahnya adalah:
   A. Pendefinisian fungsi F dan F', konstanta K dan K ' untuk masing-masing *round*.
   B. Pada setiap blok X1, X2, ...Xn lakukan langkah-langkah sebagai berikut:
      a. Bagi Xi ke dalam 16 subblok 32 bit yaitu X[k], $0 \leq k \leq 15$, menggunakan metode endian kecil, dengan mengubah susunan urutan byte, dimana paling kanan dijadikan *high order* diikuti oleh byte selanjutnya sampai dengan byte paling kiri dijadikan *low order*.
         Set A=A'=H0,
         B=B'=H1, C=C'= H2, D=D'=
         H3, E=E'= H4.

      b. Kemudian, lakukan semua operasi yang ada pada kedua rangkaian rantai paralelnya. Setiap operasinya berupa fungsi (a, b, c, d, e, X[i], s], yang mendefinisikan operasi :

T ← ((A*f*(*j*,B,C,D) X*i*[*r*(*j*)] K(j) <<*s*(*j*))E;

(A ← E); (E ← D);(D ← C<<10); (C ← B); (B ← T);

T' ← ((A' ⊞ *f*(*79-j*,B',C',D') ⊞X*i*[*r*(*j*)] ⊞K'(j) <<*s'*(*j*)) ⊞ E';

(A' ← E'); (E' ← D'); (D' ← C'<<10); (C' ← B'); (B' ← T');

Dimana f adalah nama fungsi, *buffer* a, b, c, d dan e adalah buffer untuk menampung nilai sementara, X[i] adalah notasi dari 16 subblok dan *k* adalah konstanta yang digunakan untuk masing-masing round,sedangkan <<*s* adalah operasi pergeseran bit ke kiri sebanyak s bit, serta simbol ⊞ dinotasikan sebagai *addition modulo (add mod)* $2^{32.}$

3. HASIL DAN PEMBAHASAN

3.1 Pembahasan

Proses perhitungan manual pada citra tanda tangan menggunakan algoritma *hash* RIPEMD160, dengan ekstensi jpeg, resolusi 902 x 1280*pixel*. Untuk memudahkan proses analisa maka diambil sampel dari hasil citra tanda tangan berukuran 5 x 5 *pixel*. Nilai *pixe*l citra tanda tangan (sampel) dapat dilihat pada tabel berikut:

Tabel 1 nilai RGB dari citra tanda tangan berdasarkan gambar 3.3

193	196	204	215	229
138	143	155	174	200
88	92	101	117	141
68	68	71	79	97
68	68	68	68	72

Berikut ini langkah-langkah penerapan algoritma RIPEMD160 untuk mendeteksi keaslian tanda tangan, dimana sebelumnya terlebih dahulu dilakukan pengubahan nilai *pixel* dari citra tanda tangan menjadi bilangan biner. Dalam pengujian nilai yang akan digunakan adalah 193, 196 , 204, 215, 229, 138, 143, 155, 174, 200, 88, 92, 101, 117, 141, 68, 68, 71, 79, 97, 68, 68, 68, 68, 72

Tabel 2. nilai *pixel* RGB dalam biner berdasarkan gambar 3.3

11000001	11000100	11001100	11010111	11100101
10001010	10001111	10011011	10101110	11001000
01011000	01011100	01100101	01110101	10001101
01000100	01000100	01000111	01001111	01100001
01000100	01000100	01000100	01000100	01001000

Berdasarkan rumus diatas maka perhitungan nya adalah sebagai berikut:
Step 1 Blok Kiri :
==========
A = 01100111010001010010001100000001
B = 11101111110011011010101110001001
C = 10011000101110101101110011111110
D = 00010000001100100101010001110110
E = 11000011110100101110000111110000
PermutasiKiri(Round = 0, Step = 0) = 0
maka nilai input yang digunakan -> X[0] = 11000001110001001100110011010111 K[0] = 0
S(Round = 0, Step = 0) = 11
Fungsi yang digunakan = f1
F(B,C,D) = B Xor C Xor D = 01100111010001010010001100000001
A AddModulo F(B,C,D) = 11001110100010100100011000000010
A AddModulo F(B,C,D) AddModulo X = 10010000010011110001001011011010
A AddModulo F(B,C,D) AddModulo X AddModulo K = 10010000010011110001001011011010

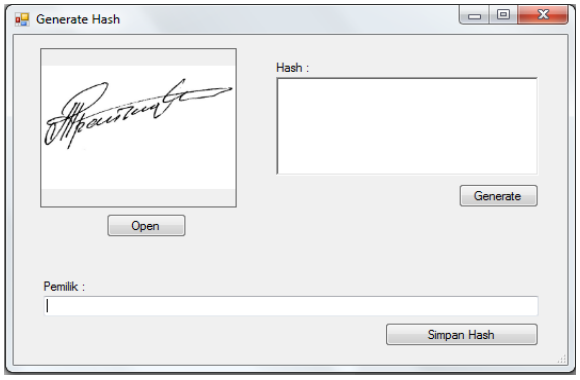
A AddModulo F(B,C,D) AddModulo X AddModulo K RotateLeft S =
01111000100101101101010010000010
A AddModulo F(B,C,D) AddModulo X AddModulo K RotateLeft S AddModulo E =
00111100011010011011011001110011
Rotasi Variabel A <- E, E <- D, D <- ROL(C,10), C <- B, B <- T A =
11000011110100101110000111110000 B = 00111100011010011011011001110011
C = 11101111110011011010101110001001
D = 11101011011100111111101001100010
E = 00010000001100100101010001110110
Step 1 Blok Kanan :
==========
A' = 01100111010001010010001100000001
B' = 11101111110011011010101110001001
C' = 10011000101110101101110011111110
D' = 00010000001100100101010001110110
E' = 11000011110100101110000111110000
PermutasiKanan(Round = 0, Step = 0) = 5
maka nilai input yang digunakan -> X[5] = 01000100010001000100010001000100
K'[0] = 1352829926
S'(Round = 0, Step = 0) = 8
Fungsi yang digunakan = f5
F(B',C',D') = B' Xor (C' Or ~D') = 00010000001100100101010001110110
A' AddModulo F(B',C',D') = 01110111011101110111011101110111
A' AddModulo F(B',C',D') AddModulo X = 10111011101110111011101110111011
A' AddModulo F(B',C',D') AddModulo X AddModulo K' = 00001100010111100100011110100010
A' AddModulo F(B',C',D') AddModulo X AddModulo K' RotateLeft S' =
01011110010001111010001000001100
A' AddModulo F(B',C',D') AddModulo X AddModulo K' RotateLeft S' AddModulo E' =
00100010000110101000001111111101
Rotasi Variabel A' <- E', E' <- D', D' <- ROL(C',10), C' <- B', B' <- T' A' =
11000011110100101110000111110000 B' = 00100010000110101000001111111101
C' = 11101111110011011010101110001001
D' = 11101011011100111111101001100010
E' = 00010000001100100101010001110110
Step 16 Blok Kanan :
==========
A' = 11100001010000010011101110001011
B' = 10100100011010101100010110010110
C' = 10000101101001001110101000110001
D' = 11000010011011110000111010001011
E' = 10110000000110010011110011001101
PermutasiKanan(Round = 4, Step = 15) = 11
maka nilai input yang digunakan -> X[11] = 00000000000000000000000000000000
K'[4] = 0
S'(Round = 4, Step = 15) = 11
Fungsi yang digunakan = f1
F(B',C',D') = B' Xor C' Xor D' = 11100011101000010010000100101100
A' AddModulo F(B',C',D') = 11000100111000100101110010111000
A' AddModulo F(B',C',D') AddModulo X = 11000100111000100101110010111000
A' AddModulo F(B',C',D') AddModulo X AddModulo K' = 11000100111000100101110010111000
A' AddModulo F(B',C',D') AddModulo X AddModulo K' RotateLeft S' =
00010010111001011100011000100111
Output Penggabungan Akhir :
==========
T = H(1) + C + D' = 331590301
H(1) = H(2) + D + E' = 1058095635
H(2) = H(3) + E + A' = 2438415222
H(3) = H(4) + A + B' = 3039469224
H(4) = H(0) + B + C' = 1058095635
H(0) = T = 331590301s
Sehingga diperoleh hasil Hash: 13C3AA9D3F11421391574376B52A9EA8826F0230

ISSN 2807-9507 (Media Online)
Vol 1, No 3, Maret 2022
Hal 110-116
https://djournals.com/jieee

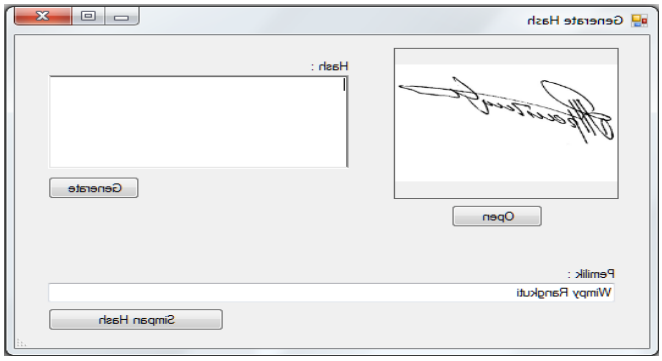
Sehingga kita mendapatkan hasil hash dari step 1 sampai ke step 16 yaitu
13C3AA9D3F11421391574376B52A9EA8826F0230
Selanjutnya adalah penentuan Riwayat step 1-6 yang dapat anda lihat pada tabel berikut
Output penggabungan akhir:
==========
T = H(1) + C + D' = 331590301
H(1) = H(2) + D + E' = 1058095635
H(2) = H(3) + E + A' = 2438415222
H(3) = H(4) + A + B' = 3039469224
H(4) = H(0) + B + C' = 1058095635
H(0) = T = 331590301
Dari tabel perhitungan diatas sesuai dengan ketentuan penggunaan metode algoritma ripemd160 yaitu nilai Hash untuk data cari pixel tanda tangan tersebut yaitu : 13C3AA9D3F11421391574376B52A9EA8826F0230

3.2 Implementasi

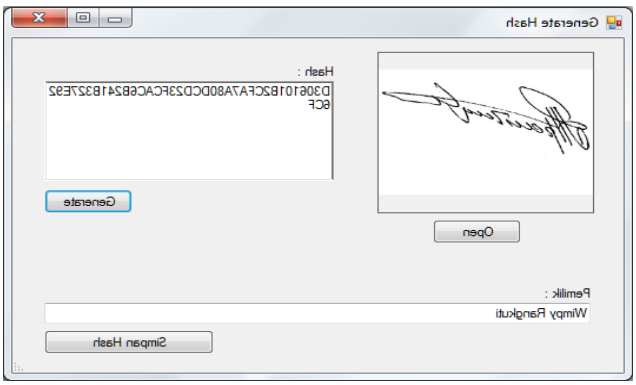
Berdasarkan contoh kasus yang sudah dikerjakan diatas maka dibuat sebuah sistem berbasis aplikasi ms visual basic 2010 . Adapun tampilan dari sistem tersebut yaitu:



Gambar 1. Tampilan Antar muka Prose Pemilihan Data Tanda Tangan

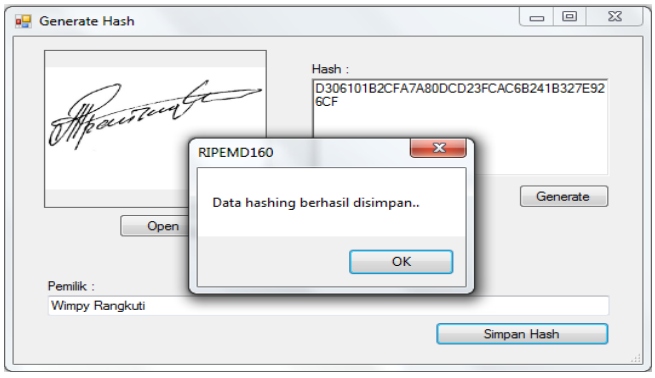


Gambar 2. Penginputan Nama Pemilik Tanda Tangan

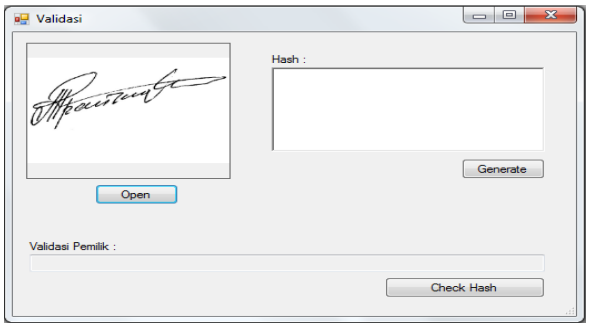


Gambar 3. Tampilan Halaman Generate Hash

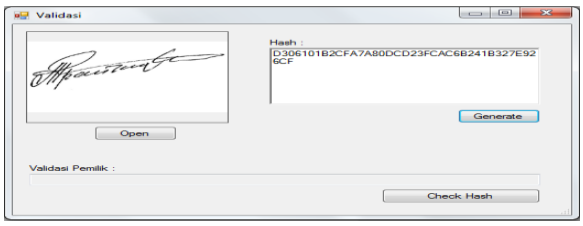
ISSN 2807-9507 (Media Online)
Vol 1, No 3, Maret 2022
Hal 110-116
https://djournals.com/jieee



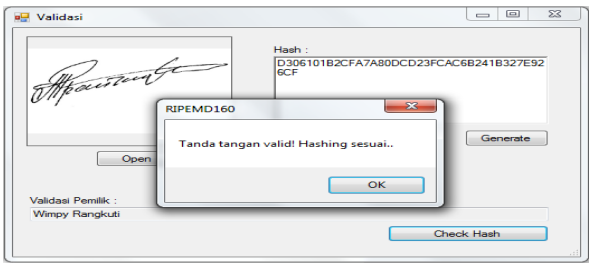
Gambar 4. Halaman Penyimpanan Hash Citra Tanda Tangan



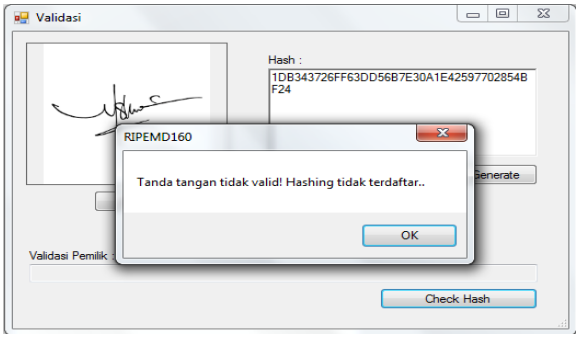
Gambar 5. Tampilan Halaman *validasi*



Gambar 6. Halaman Pengecekan Hash



Gambar 7. Halaman Pengecekan Hash Barhasil Diidentifikasi



Gambar 8. Halaman Pengecekan Hash Tidak Barhasil Diidentifikasi

4. KESIMPULAN

Berdasarkan penelitian Analisa Keaslian Tanda Tangan Dengan Menerapkan Algoritma RIPEMD160 maka dapat diambil beberapa kesimpulan yaitu: Tanda tangan dapat diberikan sebuah penanda atau sebuah informasi khusus dengan memberikan nilai Hash khusus yang hanya dimiliki oleh citra tersebut untuk memberikan semacam ID unik. Penggunaan ID unik atau Hash ini akan menjadi pemisah antara tanda tangan yang asli dengan tanda tangan yang palsu. Penerapan Hash akan degenerate dari data citra dengan menginputkan password sebagai penginisiasi data awal untuk

REFERENCES

[1] Arius, D. (2008). *Keamanan Multimedia*. Yogyakarta: Penerbit Andi.

[2] Larsson, & Moffat. (2000). *Off-Line Dictionary-Based Compression.* Kyoto: IEEE.

[3] Sudewa. (2003) Data File dan Jenisnya Dalam Komputer

[4] Ariani, R. & Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika

[5] Suarga. 2006. *Algoritma dan Pemrograman.* Andi: Yogyakarta.

[6] Mhd Arief Hasan. 2017. *Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Lancang Kuning Riau)*. JURNAL TEKNOLOGI DAN SISTEM INFORMASI - VOL. 03 NO. 02 (2017) 291-298

[7] Phie Chyan. 2017. *PENERAPAN SISTEM KRIPTOGRAFI ENKRIPSI JAMAK DAN TANDA TANGAN DIGITAL DALAM MENDUKUNG KEAMANAN INFORMASI*. Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar

[8] Sebastian Suhandinata. 2014. *Implementasi Metode Beaufort Cipher Dan Blowfish Cipher untuk Enkripsi SMS Pada Telepon Seluler Berbasis Android*. JKonferensi Nasional Sistem dan Informatika 2014; Bali, November 7-8,

[9] Mia Diana. 2018. *IOptimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS*. Jurnal Sains Komputer & Informatika (J-SAKTI) Volume (2) No.1 Maret 2018, pp. 12-22

[10] James R. Silkenat. 2014 *The American Bar Association and the Rule of La.*
SMU Law Review Volume 67 | Issue 4 Article 7

[11] Mohamad Hoseyn Sigari. 2011. Offline Handwritten Signature Identification and Verification Using Multi-Resolution Gabor Wavelet. International Journal of Biometrics and Bioinformatics (IJBB), Volume (5) : Issue (4) : 201