PENERAPAN ALGORITMA MESSAGE DIGEST 5 PADA FILE VIDEO

Rivalri Kristianto Hondro
Universitas Budi Darma
Jalan Sisingamangaraja No. 338 Simpang Limun, Kota Medan
rivalryhondro@gmail.com

ABSTRACT

Using the video document as a medium to convey information that is widely used today. Where proven in several social networking applications much content on the information in the form of video. Noting the security side of the video that is intended to secure the information in the videos cannot be accessed by unauthorized persons. In addition, the security side to ensure is rejected from the data that should be avoided-manipulation actions performed by people who are not responsible. Technical security video that can regulate the freedom or authenticity of the video cryptographic techniques. One algorithm cryptography was able to verify from the video is the message-digest algorithm 5. The results obtained from the application message digest 5 is a hash code that can be used as the value of the meter to the authenticity of the identity meta-approved video data.

Keyword: *Authentication, Cryptography, Message Digest 5 Algorithm, Video*

PENDAHULUAN

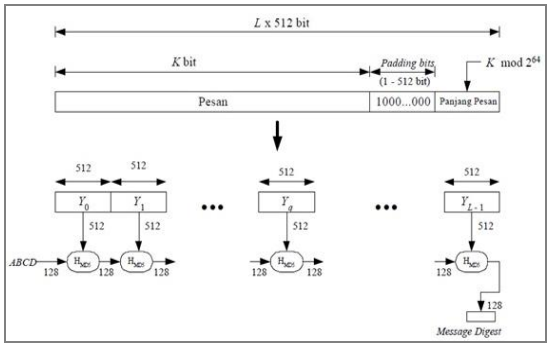
Pada jaman ini kebanyakan orang, instansi swasta dan pemerintahan menggunakan pengolahan data berbasis komputer sehingga menghasilkan representasi data dalam bentuk data digital. Bukan itu saja proses pendistribusian data juga dilakukan dengan menggunakan teknologi internet dengan memanfaatkan sejumlah aplikasi layanan pengiriman data seperti E-Mail dan aplikasi jejaringan social seperti Facebook, Twitter, Instagram, Whatsapp (Hondro, 2015). Maka dengan situasi seperti ini bisa menimbulkan beberapa tindakan penyadapan dan pemanipulasian data yang dilakukan oleh orang yang tidak berkepentingan, maka diperlukan sebuah otentikasi dokumen. Pada artikel Hondro, dkk, menjelesakan bahwa Otentikasi merupakan kegiatan verifikasi dokumen digital dimana dokumen tersebut dapat diakses oleh orang yang berhak (Zebua, et al., 2018). Selain itu dapat juga memastikan dokumen digital itu asli atau tidak.

Video adalah salah satu jenis data digital yang digunakan orang saat ini untuk menyampaikan sebuah informasi. Informasi yang ada dalam video tentunya perlu dijaga keasliannya sehingga orang berkepentingan dapat meyakinkan dirinya bahwasannya video tersebut asli (Gunawan, 2018). Cara yang tepat untuk mengatasi permasalahan tersebut yaitu dengan menerapkan teknik kriptografi jenis hash (Purwanti et al., 2013). Dimana dengan menerpkan teknik ini

maka dihasilkan sebuah kode identitas sebuah file (Hondro, et al., 2014). Algoritma kriptografi jenis hash yang dapat diterapkan dalam proses otentikasi file video yaitu algoritma message digest 5.

METODE

Algoritma message digest 5 adalah jenis fungsi hash satu arah. Fungsi hash adalah proses pemberian kode otentikasi terhadap file secara efisien yang mana akan mengubah string input dengan panjang tak terhingga menjadi string output dengan panjang tetap yang disebut nilai hash (Munir, 2019). Algoritma message digest 5 dibuat oleh seorang pakar kriptografi bernama Ronal Rivest pada tahun 1991(Waqidiyanto, 2018).



Gambar 1. Pembuatan message digest 5

Langkah-langkah pembuatan message digest 5 (Mogarala & Kabadi, 2018):

1. Menambahkan *padding bits*.
   a. Pesan ditambah dengan sejumlah *padding bits* sedemikian sehingga panjang pesan (dalam satuan *bit*) kongruen dengan 448 *modulo* 512.
   b. Jika panjang pesan 448 *bit*, tambahkan 512-bit sehingga menjadi 960 *bit*. Jadi, panjang *bit padding bits* adalah antara 1 sampai 512.
   c. *Padding bits* terdiri atas sebuah *bit* 1 dan, sisanya, yang mengikutinya, *bit* 0.
2. Menambahkan nilai panjang pesan.
   a. Pesan yang telah diberi *padding bits* selanjutnya ditambah lagi dengan 64-bit yang menyatakan panjang pesan semula.
   b. Jika panjang pesan $>2^{64}$, yang diambil adalah panjangnya dalam modulo 2^{64}. Dengan kata lain, jika panjang pesan semula adalah K *bit*, 64-bit yang ditambahkan menyatakan K *modulo* 2^{64}.
   c. Setelah ditambah dengan 64 *bit*, panjang pesan sekarang menjadi kelipatan 512 *bit.*
3. Menginisialisasi penyangga MD
   a. MD5 membutuhkan 4 buah penyangga yang masing-masing panjangnya 32 *bit*. Total panjang penyangga adalah 4 x 32 = 128 *bit*. Keempat penyangga ini menampung hasil antara dan hasil akhir.
   b. Keempat penyangga dinamai A, B, C, dan D. Setiap penyangga

dinisialisasi dengan nilai-nilai (dalam notasi *HEX*) berikut:

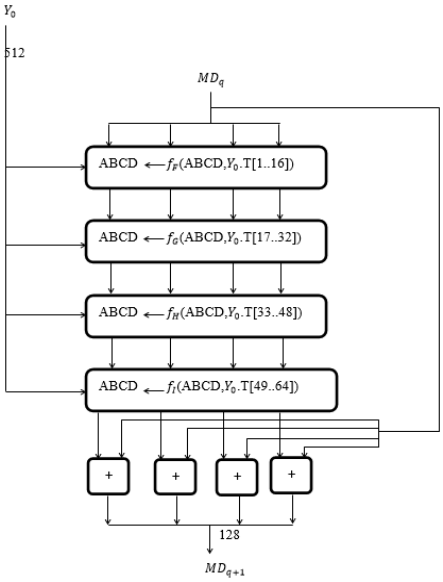
A = 01 23 45 67

B = 89 AB CD EF

C = FE DC BA 98

D = 76 54 32 10

4. Mengolah pesan dalam blok berukuran 512 *bit*.
   a. Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 *bit* (Y_0 sampai Y_{L-1}).
   b. Setiap blok 512-bit diproses bersama dengan penyangga MD menjadi keluaran 128-bit, yang disebut proses H_{MD5} seperti diperlihatkan pada Gambar 1. Proses H_{MD5} terdiri atas 4 buah putaran. Masing-masing putaran melakukan operasi dasar memakai sebuah elemen T. Jadi, setiap putaran memakai 16 elemen tabel T.
   c. Pada Gambar 1, Yq menyatakan blok 512-bit ke-q dari pesan yang telah ditambahi *padding bits* dan tambahan 64-bit nilai panjang pesan semula. MDq adalah nilai message digest 128-bit dari proses H_{MD5} ke-q. Pada awal proses, MDq berisi nilai inisialisasi penyangga MD.



Gambar 2. Pengolahan blok 512 *bit* (proses H_{SHA}).

Fungsi-fungsif_F, f_G, f_H, dan f_I masing-masing berisi 16 kali operasi dasar terhadap masukan (Weis, S. A., Sarma, S. E., Rivest, R. L., & Engels, 204 C.E.). Setiap operasi dasar menggunakan elemen tabel *T*. Operasi dasar MD5 dapat ditulis dalam bentuk persamaan pada gambar 2.1.

a ← b + CLSs (a + g (b, c, d) + X [k] + T [i])

Keterangan:

a, b, c, d : empat buah peubah penyangga 32 *bit* (berisinilaipenyangga A, B, C, D)

g : salah satu fungsi F, G, H, I

CLSs : circular *left shift* sebanyak s *bit*

X [k] : kelompok 32-bit ke-k dariblok 512-bit *message* ke-q.

Nilai k : 0 sampai 15

T [i] : elemen tabel T ke-I (32 *bit*)

+ : operasi penjumlahan modulo 232

Fungsi f_F, f_G, f_H, dan f_I adalah fungsi untuk memanipulasi masukan a, b, c, dan d dengan ukuran 32 byte Masing-masing fungsi dapat dilihat pada tabel 1

Tabel 1. Fungsi Fungsi dasar MD5

Nama	Notasi	G(b, c, d)
Ff	F (b, c, d)	(b ∧ c) ∨ (~ b ∧ d)
Fg	G (b, c, d)	(b ∧ c) ∨ (c ∧ ~ d)
fH	H (b, c, d)	b ⊕ c ⊕ d
Fi	I (b, c, d)	c ⊕ (b ∧ ~ d)

Tabel 2. Nilai T [1]

T[1]=D76A A478	T[17]=F61E 2562	T[33]=FFFA 3942	T[49]=F429 2244
T[2]=E8C7 B756	T[18]=C0B0 B340	T[34]=8771 F681	T[50]=432A FF97
T3]=242070 DB	T[19]=265E 5A51	T[35]=69D9 6122	T[51]=AB94 23A7
T[4]=C1BD CEEE	T[20]=E9B6 C7AA	T[36]=FDE5 380C	T[52]=FC93 A039
T[5]=F57C0 FAF	T[21]=D62F 105D	T[37]=A4B EEA44	T[53]= 655B59C3
T[6]=4787C 62A	T[22]=0244 1453	T[38]=4BD ECFA9	T[54]=8FOC CC92
T[7]=A8304 613	T[23]=D8A1 E681	T[39]=F6BB 4B60	T[55]=FFEF F47D
T[8]=FD469 501	T[24]=E7D3 FBCB	T[40]=BEB FBC70	T[56]=85845 DD1
T[9]=69809 8D8	T[25]=21E1 CDE6	T[41]=289B 7EC6	T[57]=6FA8 7E4F
T[10]=8B44 F7AF	T[26]=C337 07D6	T[42]=EAA 127FA	T[58]=FE2C E6E0
T[11]=FFFF 5BB1	T[27]=F4D5 0D87	T[43]=D4EF 3085	T[59]=A301 4314
T[12]=895C D7BE	T[28]=455A 14ED	T[44]=0488 D051	T]60]=4E08 11A1
T[13]=6B90 1122	T[29]=A9E3 E905	T[45]=D9D4 D039	T[61]=F753 7E82
T[14]=FD98 7193	T[30]=FCEF A3F8	T[46]=E6D B99E5	T[62]=BD3 AF235
T[15]=A679 438E	T[31]=676F 02D9	T[47]=1FA2 7CF8	T[63]=2AD7 D2BB
T[16]=49B4 0821	T[32]=8D2A 4C8A	T[48]=C4A C5665	T[64]=EB86 D391

Dari Tabel 2 dapat dilihat bahwa masing-masing fungsif_F, f_G,f_H, dan f_I melakukan 16 operasi dasar .

Misalnya notasi [abcd k s i] menyatakan operasi.

A ← b + ((a + g ((b, c, d) + X[k] + T[i]<<<s)

Dimana<<< s melambangkan dapat ditabulasikan sebagai berikut (Sadikin, 2012) :

1. Putaran 1:16 kali operasi dasar dengan g (b, c, d) = F (b, c, d)

Tabel 3. Perincian operasi pada fungsi F (b, c, d)

No.	[abcd k s i]
1.	[ABCD 0 7 1]
2.	[DABC 1 12 2]
3.	[CDAB 2 17 3]
4.	[BCDA 3 22 4]
5.	[ABCD 4 7 5]
6.	[DABC 5 12 6]
7.	[CDAB 6 17 7]
8.	[BCDA 7 22 8]
9.	[ABCD 8 7 9]
10.	[DABC 9 12 10]
11.	[CDAB 10 17 11]
12.	[BCDA 11 22 12]
13.	[ABCD 12 7 13]
14.	[DABC 13 12 14]
15.	[CDAB 14 17 15]
16.	[BCDA 15 22 16]

2. Putaran 2:16 kali operasi dasar dengan g (b, c, d) = G (b, c, d)

Tabel 4. Perincian operasi pada fungsi G (b, c, d)

No.	[abcd k s i]
17.	[ABCD 1 5 17]
18.	[DABC 6 9 18]
19.	[CDAB 11 14 19]
20.	[BCDA 0 20 20]
21.	[ABCD 5 5 21]
22.	[DABC 10 9 22]
23.	[CDAB 15 14 23]
24.	[BCDA 4 20 24]
25.	[ABCD 9 5 25]

26.	[DABC 14 9 26]
27.	[CDAB 3 14 27]
28.	[BCDA 8 20 28]
29.	[ABCD 13 5 29]
30.	[DABC 2 9 30]
31.	[CDAB 7 14 31]
32.	[BCDA 12 20 32]

3. Putaran 3:16 kali operasi dasar dengan g (b, c, d) = H (b, c, d)

Tabel 5. Perincian operasi pada fungsi H (b, c, d)

No.	[abcd k s i]
33.	[ABCD 5 4 33]
34.	[DABC 8 11 34]
35.	[CDAB 11 16 35]
36.	[BCDA 14 23 36]
37.	[ABCD 1 4 37]
38.	[DABC 4 11 38]
39.	[CDAB 7 16 39]
40.	[BCDA 10 23 40]
41.	[ABCD 13 4 41]
42.	[DABC 0 11 42]
43.	[CDAB 3 16 43]
44.	[BCDA 6 23 44]
45.	[ABCD 9 4 45]
46.	[DABC 12 11 46]
47.	[CDAB 15 16 47]
48.	[BCDA 2 23 48]

4. Putaran 4:16 kali operasi dasar dengan g (b, c, d) = I (b, c, d)

Tabel 6. Perincian operasi pada fungsi I (b, c, d)

No	[abcd k s i]
49	[ABCD 0 6 49]
50	[DABC 7 10 50]
51	[CDAB 14 15 51]
52	[BCDA 5 21 52]
53	[ABCD 12 6 53]
54	[DABC 3 10 54]
55	[CDAB 10 15 55]
56	[BCDA 1 21 56]
57	[ABCD 8 6 57]
58	[DABC 15 10 58]
59	[CDAB 6 15 59]
60	[BCDA 13 21 60]
61	[ABCD 4 6 61]
62	[DABC 11 10 62]
63	[CDAB 2 15 63]
64	[BCDA 9 21 64]

Setelah putaran keempat, a, b, c, dan d ditambah kan ke A, B, C dan D. Selanjutnya, algoritma memproses blog data berikutnya (Y_{q+1}). Keluaran akhir dari algoritma MD5 adalah hasil penyambungan bit di A, B, C, dan D.

Dari uraian di atas, secara umum fungsi hash MD5 dapat ditulis dalam persamaan matematis berikut:

$MD_0 = IV$

$MD_{q+1} = MDq + fI (Yq + fH (YQ + fF (Yq + MDq)))$

$MD = MDL - 1$

Keterangan:

IV : initial vector dari penyangga ABCD, yang dilakukan pada proses inisialisai penyangga

Yq : blok pesan berukuran 512-bit ke-q

L : jumlah blok pesan

MD : nilai akhir *message digest*

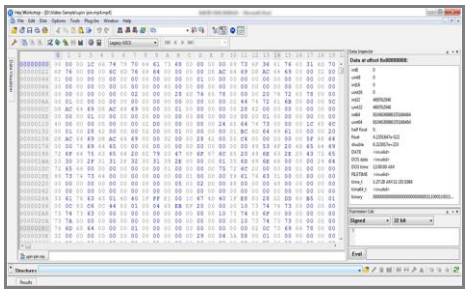
HASIl DAN PEMBAHASAN

1. Otentikasi File Video

   Otentikasi file video dengan ekstensi file MP4 dilakukan dengan cara melakukan proses ekstrasi nilai terlebih dahulu, dengan menggunakan bantuan aplikasi *Hex Workshop*. Seperti pada tampilan berikut ini

Gambar 3 Skema Ekstrasi Nilai Hexadecimal

Hasil proses penggunaan aplikasi *hex workshop*.



Gambar 4. Tampilan form Hex Workshop

2. Penerapan Algoritma Message Digest 5

   Otentikasi terhadap file video dilakukan dengan menerapkan algoritma message digest 5. Dari gambar 4 data video di atas diambil sebanyak 17-byte untuk sebagai sampel data penerapan MD5, yaitu: 000001C66747970646173680000000069

   a. Menambah Padding Bits (Panjang Pesan)

      Dari nilai hexadecimal sebagai sampel yang digunakan berjumlah 136-bit, agar pesan mencukupi 512-bit maka dilakukan penambahan bit, dimana bit pertama bernilai 1 (satu) selanjutnya diikuti bit 0 (nol). Delapan bit terakhir adalah nilai jumlah bit pesan yaitu 136 = 10001000. Berikut hasil padding bit:

Tabel 7. Tabel Padding

00000000	00000000	00000000	00011100	01100110	01110100	01111001
01110000	01100100	01100001	01110011	01101000	00000000	00000000
00000000	00000000	01101001	**1**0000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
11111000						

b. Inisialisasi Penyangga Nilai *Hash Message Diggest* (*MD*)

Fungsi f_F, f_G, f_H, dan f_I adalah fungsi untuk memanipulasi masukan a, b, c, dan d dengan ukuran 32 *bit*, dan lihat table fungsi pada table 1

Penyelesaian fungsi fungsi dasar MD5:

1. F (b, c, d) = (b ∧ c) ∨ (~ b ∧ d)

   = (89ABCDEF ∧ FEDCBA98) ∨ (~89ABCDEF ∧ 76543210)

a. (b ∧ c) = (89ABCDEF ∧ FEDCBA98)

b. (~ b ∧ d) = (~89ABCDEF ∧ 76543210)

c. (b ∧ c) ∨ (~ b ∧ d)

Maka konversi nilai biner diatas ke Nilai Hexadecimal (Fungsi "F") = FEDCBA98. Lakukan hal sama dengan fungsi G, H, dan I.

2. G (b, c, d) = (b ∧ d) ∨ (c ∧ ~ d) Hasil Nilai Hexadecimal (Fungsi "G") = 88888888
3. H (b, c, d) = b ⊕ c ⊕ d Hasil Nilai Hexadecimal (Fungsi "H") = 01234567
4. I (b, c, d) = c □ (b ∧ ~ d) Hasil Nilai Hexadecimal (Fungsi "I") = 77777777

c. Pengolahan Pesan Dalam Blok Berukuran 512 *Bit* (*Parsing*) Bit pesan bagi menjadi 16 blok, dengan jumlah bit pada masing-masing blok berjumlah 32 bit

Selanjutnya lakukan perhitungan:

1. Putaran 1 - 16: menggunakan 16 kali operasi dasar dengan g (b, c, d) = F (b, c, d)

   Putaran Ke - 1:
   A □ 89ABCDEF + ((01234567 + FEDCBA98 + 0000001C + D76AA478)<<<7)

   01234567 = 0000 0001 0010 0011 0100 0101 0110 0111
   FEDCBA98 = 1111 1110 1101 1100 1011 1010 1001 1000

   M_0 = 0000 0000 0000 0000 0000 0000 0001 1100 D76AA478 = 1101 0111 0110 1010 1010 0100 0111 1000

Di XOR maka hasilnya
Hasil =11101 0111 0110 1010 1010 0100 1001 0011

Tulisan 1 bercetak tebal dibuang menjadi angka **1101 011**1 0110 1010 1010 0100 1001 0011 <<<7
Tulisan bercetak tebal disebelah kiri digeser ke kanan
1011 0101 0101 0010 0100 1001 1110 1011

89ABCDEF = 1000 1001 1010 1011 1100 1101 1110 1111 Hasil A = **1**0011 1110 1111 1110 0001 0111 1101 1010
Nilai Hex = 3EFE17DA

Lakukan proses untuk 2 s/d 16 seperti proses putaran ke 1 dan ke 2, demikian juga dengan putaran ke 17 s/d 32, putaran 33 s/d 48, dan putaran 49 s/d 64, dimana Putaran 17 - 32: menggunakan 16 kali operasi dasar dengan g (b, c, d) = G (b, c, Putaran 33 - 48: menggunakan 16 kali operasi dasar dengan g (b, c, d) = H (b, c, Putaran 49 - 64: menggunakan 16 kali operasi dasar dengan g (b, c, d) = I (b, c, Berikut hasil perputaran 1 s/d 64 pada tabel 8:

Int	A	B	C	D
T_0	01234567	89ABCDEF	FEDCBA98	76543210
T_1	3EFE17DA	01234567	89ABCDEF	FEDCBA98
T_2	4CB822E2	3EFE17DA	01234567	89ABCDEF
T_3	5230DEF2	4CB822E2	3EFE17DA	01234567
T_4	451C3D62	5230DEF2	4CB822E2	3EFE17DA
T_5	07B3A51E	451C3D62	5230DEF2	4CB822E2
T_6	064E5267	07B3A51E	451C3D62	5230DEF2
T_7	15D11E4F	064E5267	07B3A51E	451C3D62
T_8	C9EB1F94	15D11E4F	064E5267	07B3A51E
T_9	C9F839A3	C9EB1F94	15D11E4F	064E5267
T_{10}	D92636A3	C9F839A3	C9EB1F94	15D11E4F
T_{11}	7521EDEE	D92636A3	C9F839A3	C9EB1F94
T_{12}	790E2524	7521EDEE	D92636A3	C9F839A3
T_{13}	51B45EA4	790E2524	7521EDEE	D92636A3
T_{14}	10C4FDC8	51B45EA4	790E2524	7521EDEE
T_{15}	10C71AE1	10C4FDC8	51B45EA4	790E2524
T_{16}	CFBE3AF1	10C71AE1	10C4FDC8	51B45EA4
T_{17}	5179662B	CFBE3AF1	10C71AE1	10C4FDC8
T_{18}	62AE2C82	5179662B	CFBE3AF1	10C71AE1
T_{19}	13BBF9F1	62AE2C82	5179662B	CFBE3AF1
T_{20}	E5030418	13BBF9F1	62AE2C82	5179662B
T_{21}	8707977A	E5030418	13BBF9F1	62AE2C82
T_{22}	69705306	8707977A	E5030418	13BBF9F1
T_{23}	E705E682	69705306	8707977A	E5030418
T_{24}	25597DEB	E705E682	69705306	8707977A
T_{25}	FD5F48A4	25597DEB	E705E682	69705306
T_{26}	4F575888	FD5F48A4	25597DEB	E705E682
T_{27}	C2895D6B	4F575888	FD5F48A4	25597DEB
T_{28}	B178BE4D	C2895D6B	4F575888	FD5F48A4
T_{29}	FBA2AC75	B178BE4D	C2895D6B	4F575888
T_{30}	83766DC4	FBA2AC75	B178BE4D	C2895D6B
T_{31}	3DDE0A35	83766DC4	FBA2AC75	B178BE4D
T_{32}	313D3B50	3DDE0A35	83766DC4	FBA2AC75
T_{33}	ADB7EEEF	313D3B50	3DDE0A35	83766DC4
T_{34}	CDB64A3C	ADB7EEEF	313D3B50	3DDE0A35

T_{35}	759C3 A0E	CDB64 A3C	ADB7E EEF	313D3 B50
T_{36}	F6AB E3D0	759C3 A0E	CDB64 A3C	ADB7 EEEF
T_{37}	5D0A B60F	F6ABE 3D0	759C3 A0E	CDB64 A3C
T_{38}	B47F8 BAC	5D0AB 60F	F6ABE 3D0	759C3 A0E
T_{39}	5FDA C6F0	B47F8 BAC	5D0AB 60F	F6ABE 3D0
T_{40}	290C5 112	5FDAC 6F0	B47F8 BAC	5D0A B60F
T_{41}	37CC6 731	290C51 12	5FDAC 6F0	B47F8 BAC
T_{42}	C742F 556	37CC6 731	290C51 12	5FDA C6F0
T_{43}	44FFA 524	C742F5 56	37CC6 731	290C5 112
T_{44}	732F3 542	44FFA 524	C742F5 56	37CC6 731
T_{45}	4B617 E6C	732F35 42	44FFA 524	C742F 556
T_{46}	9AD16 D38	4B617E 6C	732F35 42	44FFA 524
T_{47}	9269E FD8	9AD16 D38	4B617E 6C	732F35 42
T_{48}	D7417 819	9269EF D8	9AD16 D38	4B617 E6C
T_{49}	BAA3 9D8A	D74178 19	9269EF D8	9AD16 D38
T_{50}	A09D A4DD	BAA39 D8A	D74178 19	9269E FD8
T_{51}	F9EE6 006	A09DA 4DD	BAA39 D8A	D7417 819
T_{52}	2C9A7 3BA	F9EE60 06	A09DA 4DD	BAA3 9D8A
T_{53}	073175 66	2C9A7 3BA	F9EE60 06	A09D A4DD
T_{54}	27D18 E0D	073175 66	2C9A7 3BA	F9EE6 006
T_{55}	E2598 A34	27D18 E0D	073175 66	2C9A7 3BA
T_{56}	0D986 061	E2598 A34	27D18 E0D	073175 66
T_{57}	9A7A9 969	0D9860 61	E2598 A34	27D18 E0D
T_{58}	A83E A7CA	9A7A9 969	0D9860 61	E2598 A34
T_{59}	89A4D BBC	A83EA 7CA	9A7A9 969	0D986 061
T_{60}	59A4A 248	89A4D BBC	A83EA 7CA	9A7A9 969
T_{61}	653AA 5F4	59A4A 248	89A4D BBC	A83EA 7CA
T_{62}	E06BD AC6	653AA 5F4	59A4A 248	89A4D BBC
T_{63}	8AB45 1D9	E06BD AC6	653AA 5F4	59A4A 248

Setelah putaran ke 63 a,b,c, dan d ditambahkan ke A,B,C, dan D :

T_{63}	8AB45 1D9	E06BD AC6	653A A5F4	59A4 A248
A, B, C, D	012345 67	89ABC DEF	FEDC BA98	7654 3210
Hasil	899714 BE	69C01 729	9BE61 F6C	2FF0 9058

Penyelesaian:

a. 8AB451D9 ⊕ 01234567 = 899714BE
b. E06BDAC6 ⊕ 89ABCDEF = 69C01729
c. 653AA5F4 ⊕ FEDCBA98 = 9BE61F6C
d. 59A4A248 ⊕ 76543210 = 2FF09058

Jadi, hasil hash yang didapat dari file video Aku.mp4 adalah: **899714BE69C017299BE61F6C2FF09058**.

Maka dengan kode hash tersebut dapat dijadikan sebagai identitas asli file video Aku.mp4.

3. Hasil Pengujian

Berikut hasil pengujian otentikasi file video dengan kasus perubahan yang berbeda-beda.

Tabel 9. Hasil Pengujian

No	Spesifikasi File Video Asli	Kode Hash File Video Asli	Perubahan yang dilakukan	Spesifikasi File Video Palsu	Kode Hash File Video Asli
1	Nama File: Aku Ukuran: 1,3 Mb Durasi: 10:00 Jenis: *.mp4	899714BE69C017299BE61F6C2FF09058	Memotong durasi video	Nama File: Aku Ukuran: 1 Mb Durasi: 09:55 Jenis: *.mp4	A4E40F15AE542AE32B7234506 11A5CCB
2.	Nama File: Video 2 Ukuran: 2 Mb Durasi	0C00D49F80EFA785DDCAD01A0FFFF700	Menambah kontras video	Nama File: Video 2 Ukuran: 2 Mb Durasi	BA2DDA481FDD08EDDC8AF0EA00A1FCA7

	: 12:57 Jenis: *.mp4			: 13:01 Jenis: *.mp4	
3.	Nama File: Sosial Ukuran: 820 Kb Durasi : 01:20 Jenis: *.mp4	DD0AC7F9C00E1AAF5CD4AE0DAC0ABB2E0	Menambah watermark didalam video	Nama File: Sosial Ukuran: 998 Kb Durasi : 01:20 Jenis: *.mp4	0C00D49F80EFA785DDCAD01A0FFEE700

KESIMPULAN

Penerapan message digest 5 untuk otentikasi file video dapat menghasilkan kode hash yang dapat digunakan sebagai kode penanda identitas sebuah file video. Selain itu juga kode hash tersebut dapat dijadikan sebagai penanda apakah file video tersebut masih asli atau tidak.

DAFTAR PUSTAKA

Aumasson, J. P., Neves, S., Wilcox-O'Hearn, Z., & Winnerlein, C. (2013). BLAKE2: Simpler, smaller, fast as MD5. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-38980-1_8

Gunawan, I. (2018). Penggunaan Algoritma Kriptografi Steganografi Least Significant Bit Untuk Pengamanan Pesan Teks dan Data Video. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*. https://doi.org/10.30645/j-sakti.v2i1.48

Hayati, N. (2017). Implementasi Algoritma RC4A dan MD5 untuk Menjamin Confidentiality dan Integrity pada File Teks. *Jurnal & Penelitian Teknik Informatika*.

Hondro, R. K. (2015). Aplikasi Enkripsi Dan Dekripsi SMS Dengan Algoritma Zig Zag Cipher Pada Mobile Phone Berbasis Android. *Pelita Informatika Budi Darma*, *3*(10), 122–127. https://osf.io/preprints/inarxiv/a2dwn/download

Hondro, R. K. (2020a). *Analisis Algoritma CLEFIA 128 Bit Jenis Block Cipher Untuk Pengamanan Teks*. *1*(2), 35–38.

Hondro, R. K. (2020b). *Modifikasi Platform Kunci Algoritma Playfair Untuk Meningkatkan Nilai Confusion Pada Ciphertext*. *1*(2), 76–82.

Hondro, R. K., & Nurcahyo, G. W. (2014). ANALISIS DAN PERANCANGAN SISTEM YANG MENERAPKAN ALGORITMA TRIANGLE CHAIN CIPHER (TCC) UNTUK ENKRIPSI RECORD TABEL DATABASE. *Jurnal Teknologi Informasi Dan Komputer*, *3*(2), 118–127.

Mogarala, A. G., & Kabadi, M. G. S. (2018). Proficient hamming weight based RSA-MD5 security for data storage in multi cloud environment. *International Journal of Intelligent Engineering and Systems*. https://doi.org/10.22266/IJIES2018.0430.24

Munir, R. (2019). Kriptografi. In *2*.

Purwanti, K., Hamdani, & Septiarini, A. (2013). Kriptografi Pada Video Menggunakan Metode Transposisi. *Jurnal Informatika Mulawarman*.

Sadikin, R. (2012). *Kriptografi Untuk Keamanan Jaringan* (I). ANDI OFFSET.

Sibyan, H. (2017). IMPLEMENTASI ENKRIPSI BASIS DATA DENGAN ALGORITMA MD5 (MESSAGE DIGEST ALGORITHM 5) DAN VIGENERE CIPHER. *Ppkm I*.

Turner, S., & Chen, L. (2011). Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms. *RFC*.

Waqidiyanto, A. (2018). No Proteksi URL Dengan Algoritma Md5 Dan Base 64 Untuk Pengamanan Website. *Doctoral Dissertation*.

Weis, S. A., Sarma, S. E., Rivest, R. L., & Engels, D. W. (204 C.E.). *Security and privacy aspects of low-cost radio frequency identification systems. In Security in pervasive computing* (Heidelberg (ed.)). Springer.

Zebua, T., Hondro, R. K., & Ndruru, E. (2018). Message Security on Chat App based on Massey Omura Algorithm. *IJISTECH (International Journal Of Information System & Technology)*, *1*(2), 16–23. https://doi.org/10.30645/ijistech.v1i2.11

Zheng, X., & Jin, J. (2012). Research for the application and safety of MD5 algorithm in password authentication. *Proceedings - 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012*. https://doi.org/10.1109/FSKD.2012.6234010