Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 2, No 2, February 2023
Hal 81-88
https://ejurnal.seminar-id.com/index.php/bulletinds

Penerapan Algoritma SHA-384 Pada Aplikasi Duplicate Video Scanner

Muhammad Hanafiah

Fakultas Ilmu Komputer Dan Teknologi Informasi, Teknik Informatika, Universitas Budi Darma Medan Indonesia
Email: hanafi.kapja@gmail.com

Abstrak−Kriptografi digunakan untuk menjaga keamanan dari suatu isi atau informasi yang bersifat pribadi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah disandi. Namun, seiring dengan perkembangan teknologi komputer yang pesat, tantangan terhadap keamanan informasi juga semakin meningkat. Manipulasi video merupakan salah satu bentuk ancaman serius dalam dunia digital saat ini. Manipulasi video adalah fenomena yang semakin umum di mana individu dapat dengan mudah memanipulasi video dengan tujuan yang merugikan atau membingungkan orang lain. Salah satu jenis manipulasi video yang paling berbahaya adalah duplikasi video scanner, di mana video palsu yang dibuat menyerupai video asli dalam banyak aspek, termasuk warna, bentuk, objek, dan konten informasi, tanpa perubahan yang signifikan dari pihak lain. Ini bisa mengelabui banyak penonton dan menimbulkan kerugian yang signifikan. Dalam penelitian ini, penulis mengusulkan penggunaan algoritma SHA-384 sebagai metode untuk mendeteksi duplikasi video *scanner*. Algoritma ini digunakan untuk menghasilkan tanda tangan unik atau hash dari data video, yang kemudian digunakan untuk memverifikasi keaslian video tersebut. Hasil penelitian ini memiliki potensi besar dalam mengatasi masalah serius yang terkait dengan manipulasi video. Kemampuan untuk membedakan video asli dari video palsu memiliki dampak yang signifikan dalam berbagai bidang, termasuk hukum, keamanan, dan integritas informasi. Penelitian ini memberikan solusi yang efektif untuk memastikan keaslian video dan dapat membantu dalam melawan penipuan dan manipulasi yang semakin banyak terjadi dalam lingkungan digital yang kompleks.

Kata Kunci: *Duplicate* Video *Scanner*, Algoritma SHA-384

Abstract−Cryptography is used to maintain the security of private information from anyone except those with the authority or secret keys to access encrypted information. However, with the rapid advancement of computer technology, challenges to information security are increasing. Video manipulation is one of the serious threats in the digital world today. Video manipulation is a common phenomenon where individuals can easily manipulate videos with harmful intent or to confuse others. One of the most dangerous types of video manipulation is the duplicate video scanner, in which a fake video is created to resemble the original video in many aspects, including color, shape, objects, and content, without significant changes from other parties. This can deceive many viewers and cause significant losses. In this research, the author proposes the use of the SHA-384 algorithm as a method to detect duplicate video scanners. This algorithm is used to generate a unique signature or hash from video data, which is then used to verify the authenticity of the video. The results of this research have great potential in addressing serious issues related to video manipulation. The ability to distinguish real videos from fake ones has a significant impact in various fields, including law, security, and information integrity. This research provides an effective solution to ensure video authenticity and can help combat the increasing fraud and manipulation in a complex digital environment.

Keywords: Duplicate Video Scanner, SHA-384 Algorithm

1. PENDAHULUAN

Perkembangan pemanfaatan teknologi informasi dalam membantu pekerjaan manusia berbagai jenis yang melibatkan langsung pada komputer sebagai medianya, perkembangan dalam pertukaran informasi menjadi lebih signifikan. Pada awalnya, manusia bertukar informasi dengan berbicara secara langsung atau mengirim pesan tulisan berupa surat. Kemudian, muncul media video yang memungkinkan penyajian informasi dengan cara yang lebih menarik dan estetis. Dalam era internet saat ini, pertukaran informasi semakin pesat dan mudah, terutama melalui email. Namun, pengamanan informasi menjadi semakin penting karena risiko penyalahgunaan dan perubahan informasi oleh pihak yang tidak bertanggung jawab.

Pemalsuan video adalah masalah yang kerap terjadi, terutama dalam kasus pembajakan film melalui unduhan ilegal dan penyebaran DVD bajakan. Dampaknya cukup signifikan, terutama pada industri perfilman, termasuk di Indonesia. Pembajakan film melibatkan tindakan ilegal perbanyakan dan penyebaran film tanpa izin dari pemilik hak cipta. Lebih buruk lagi, seringkali video yang dibajak dapat diedit dan dipalsukan dengan mudah, mengubah isi pesan atau informasinya, yang dapat merusak reputasi karya aslinya[1].

Untuk mengatasi permasalahan ini, diperlukan teknologi yang dapat memastikan integritas video. Algoritma SHA-384 adalah solusi yang diadopsi dalam berbagai konteks. Algoritma ini dikembangkan oleh *National Institute of Standards and Technology* (NIST) dan *National Security Agency* (NSA) dan digunakan sebagai komponen dalam *Digital Signature Standard* (DSS). Dalam konteks penggunaan algoritma SHA, pesan dengan panjang apa pun yang kurang dari 264 bit akan menghasilkan keluaran berupa 160 bit yang dikenal sebagai message digest[2]–[4]. Hal ini memungkinkan verifikasi integritas data dengan cara yang andal dan efisien, menjadikannya instrumen yang berpotensi efektif dalam memerangi pemalsuan video. Dengan menerapkan teknologi seperti SHA-384, perusahaan film dan pemilik hak cipta dapat lebih efektif melindungi karya-karya mereka dari pembajakan dan pemalsuan.

Penelitian yang dijadikan acuan dalam penelitian ini mencakup dua karya utama. Penelitian pertama, yang dilakukan oleh Chippy James pada tahun 2014, berkaitan dengan implementasi pengamanan data pada basis data menggunakan teknik kriptografi SHA-384. Fokus utama penelitian ini adalah menemukan metode untuk memastikan bahwa data aman dengan memanfaatkan SHA-384, sambil memberikan kemudahan bagi pemilik data dalam mengamankan informasi mereka tanpa perlu memahami *query* yang harus dijalankan. Penggunaan SHA-384 adalah untuk

menciptakan nilai hash yang unik untuk melindungi data dengan tingkat keamanan yang tinggi[5]. Penelitian kedua, yang dilakukan oleh H. B. Pethe pada tahun 2016, membahas peran fungsi hash seperti MD-5 dan SHA dalam menjaga integritas file. Fungsi hash digunakan untuk memeriksa apakah data telah mengalami perubahan atau masih dalam keadaan asli saat dikirim. Ini adalah aspek penting dalam pengamanan data dan memastikan bahwa data tidak dimanipulasi selama proses transmisi atau penyimpanan. Dengan memanfaatkan kedua penelitian ini, penelitian Anda dapat membangun landasan yang kuat dalam pengamanan data dan pemeliharaan integritas file[6].

Berdasarkan pemahaman latar belakang di atas, penulis memilih judul penelitian yang sesuai, yaitu Penerapan Algoritma SHA-384 dalam Aplikasi Pemindai Video Duplikat. Penggunaan algoritma SHA-384 dalam konteks pemindai video duplikat. Yang bertujuan untuk mengembangkan dan menerapkan algoritma tersebut dalam sebuah aplikasi yang dapat mengidentifikasi duplikasi video. Dengan demikian, penelitian ini akan mencoba mengatasi masalah pemalsuan video dan pembajakan yang sering terjadi dengan menggunakan teknologi yang andal dan kuat seperti SHA-384.

2. METODOLOGI PENELITIAN

2.1 KriptoGrafi

Kriptografi, sebuah ilmu yang berasal dari kata Yunani "*cryptos*" dan "*graphia*," bertujuan memahami cara menjaga pesan yang dikirim secara aman. Ini adalah bagian dari matematika yang dikenal sebagai kriptologi dan terfokus pada menjaga kerahasiaan informasi agar tidak dapat diakses oleh pihak yang tidak sah[7]. Perancang algoritma kriptografi dikenal sebagai kriptografer, sementara kriptanalis, di sisi lain, berusaha untuk membuka cipherteks menjadi plainteks tanpa memiliki kunci yang benar. Persamaan antara kriptanalis dan kriptografer adalah keduanya berperan dalam menerjemahkan cipherteks menjadi plainteks, tetapi perbedaannya terletak pada legitimasi; kriptanalis beroperasi tanpa izin, sedangkan kriptografer bekerja atas legitimasi pengirim atau penerima. Kriptografi adalah studi teknik matematis yang berkaitan dengan keamanan sistem informasi, mencakup kerahasiaan, integritas data, otentikasi, dan mencegah penyangkalan. Keempat aspek ini adalah tujuan utama dalam sistem kriptografi[8]. Kerahasiaan menjaga pesan tetap rahasia, integritas melindungi data dari perubahan yang tidak sah, otentikasi mengidentifikasi pihak-pihak yang terlibat, dan nirpenyangkalan mencegah penyangkalan aksi sebelumnya. Dalam prosesnya, terdapat istilah seperti *plaintext* (pesan asli), *ciphertext* (pesan yang telah disandikan), *cipher* (algoritma untuk menyandikan), enkripsi (proses penyandian), dan dekripsi (proses pengembalian plaintext). Kriptografi sistem adalah fasilitas atau algoritma yang digunakan untuk mengonversi *plaintext* menjadi *ciphertext* dan sebaliknya, dengan tujuan mengamankan sistem informasi[9].

2.2 Fungsi *Hash*

Fungsi Hash sangat berguna dan muncul di hampir semua aplikasi keamanan informasi, tidak hanya di dunia kriptografi saja. Aplikasi praktis mencakup pemeriksaan integritas pesan, *fingerprint* digital, otentikasi, dan berbagai aplikasi keamanan informasi lainnya memakai hashfunction[10]. Fungsi Hash berhubungan dengan keamanan data. Fungsi *Hash* juga dapat digunakan untuk proses autentikasi dan integritas data. Fungsi hash secara efisien akan mengubah *string input* dengan panjang tak terhingga menjadi stringoutputdengan panjang tetap yang disebut nilai hash[11]. Fungsi hash adalah fungsi matematis yang mengubah nilai input numerik menjadi nilai numerik yang terkompresi. Bertujuan mengkompresi nilai numerik yang diinputkan. Inputan fungsi *hash* mempunyai panjang yang beragam, namun outputan nilai hash akan selalu mempunyai panjang yang tetap. Nilai yang dikembalikan oleh fungsi *hash* disebut *message digest* atau hanya nilai hash. Prinsip utama dari fungsi hash yaitu tidak akan mungkin bisa membuat pesan (message) M' yang berhubungan dengan kode *Hash* h(M'), sama dengan *message* M:h(M')#h(M). Kode *hash* direpresentasikan dengan n *bits*, sehingga terdapat kemungkinan 2n – 1 kode *Hash*[12].

2.3 Video

Video adalah teknologi untuk menangkap, merekam. Memproses, mentransmisikan dan menata ulang gambar bergerak, biasanya menggunakan sinyal elektronik, atau media digital. Video juga merupakan sebagai gabungan gamba- gambar yang dibaca berurutan dalam suatu waktu dengan kecepantan tertentu. Tipe file dapat terbentuk dari aplikasi presentasi video ini seperti MP4. Sehungga dapat menggabungkan tipe file berbentuk suara beserta gambar yang di render ke bentuk video. Dalam proses pengambilan gambar memerlukan sebuah alat seperti *camera* digital, *camera handphone* atau software seperti sreencast-o-matic, dll[13].

2.4 Algoritma SHA-384

Standard Hash adalah *Secure Hash Standar (SHS)* dengan *SHA* sebagai algoritmanya. *Secure Hash Standard* (SHS) merupakan sebuah standar, sedangkan *Secure Hash Algorithm* (SHA) adalah algoritma yang digunakan dalam implementasinya[14]. Dalam SHA, saat pesan dengan panjang apa pun yang kurang dari 264 bit dimasukkan, algoritma ini menghasilkan keluaran berupa message digest sepanjang 160 bit. *Message digest* ini adalah representasi hasil pemrosesan pesan yang digunakan untuk memastikan integritas dan autentikasi data. SHA memiliki peran penting dalam berbagai aplikasi keamanan, termasuk untuk memverifikasi integritas data dan memastikan pesan atau data yang dikirim tidak mengalami perubahan yang tidak sah selama proses pengiriman[15].

Langkah-langkah pembuatan *message digest* secara garis besar adalah menambahkan *padding bits*, menambahkan nilai panjang pesan semula, Parsing Pesan, *Initial Hash Value.*

a. Menambahkan *padding bits.*
   1. Pesan ditambah dengan sejumlah *padding bits* sedemikian sehingga panjang pesan (dalam satuan *bit*) kongruen dengan 896 modulo 1024.
   2. JIka panjang pesan 33 *bit*, tambahkan 863 bit sehingga menjadi 896 *bit.* Jadi, panjang *bit padding bits* adalah antara 1 sampai 896.
   3. *Padding bits* terdiri atas sebuah *bit* 1 dan, sisanya, yang mengikutinya, *bit* 0.

b. Menambahkan nilai panjang pesan
   1. Pesan yang telah diberi *padding bits* selanjutnya ditambah lagi sehingga panjang pesan menjadi 1024 agar seimbang.

c. Parsing Pesan

d. *nitial Hash Value*

   Untuk SHA-384, nilai hash awal, H (0),

   $H_0^{(0)}$ = cbbb9d5dc1059ed8

   $H_1^{(0)}$ = 629a292a367cd507

   $H_2^{(0)}$ = 9159015a3070dd17

   $H_3^{(0)}$ = 152fecd8f70e5939

   $H_4^{(0)}$ = 67332667ffc00b31

   $H_5^{(0)}$ = 8eb44a8768581511

   $H_6^{(0)}$ = db0c2e0d64f98fa7

   $H_7^{(0)}$ = 47b5481dbefa4fa4.

e. Penjawalan Pesan

   $\text{SHA 384} = \text{Wt}\int_{S1}^{Mt\ (i)}(\text{Wt}-2)\ +\ (\text{Wt}-7)\ +\ S0(384)\ ((\text{Wt}-15)\ +\ (\text{Wt}-16)$

   $S1(384)(Wt-2) = \big((Wt-2)ROTR\ 1\big) + \big((Wt-2)ROTR\ 8\big) + \ ((Wt-2)\ SHR\ 7)$

   $S0(384)(Wt-15) = \big((Wt-15)ROTR\ 19\big) + \big((Wt-15)ROTR\ 61\big) + \ ((Wt-16)\ SHR\ 6)$

f. Mengkompresi Pesan

   $\text{Ch(X, Y, Z)} = \ (\text{Z}\wedge\text{Y}) + \ (\text{X}\wedge\text{Z})$

   $\text{Maj (x, y, z)} = \ (\text{x}\wedge\text{y}) + \ (\text{x}\wedge\text{z}) + (\text{y}\wedge\text{z})$

   $$\sum_0^{\{384)}(x) \ = \ ROTR28\ (X) + ROTR34(X) + ROTR(39)(X)$$

   $$\sum_1^{\{384)}(x) \ = \ ROTR14\ (X) + ROTR18(X) + ROTR(43)(X)$$

   $$T1 = h \ + \sum_1^{(384)}(e) \ + \ Ch(e,f,g) \ + \ Kt(384) + Wt$$

   $$T2 = \ \sum_0^{(384)}(a) \ + \ Maj(a,b,c)$$

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Masalah yang dihadapi ketika akan melakukan pencarian *file* video yang duplikat atau ganda pada sebuah media penyimpanan membutuhkan ketelitian dan ingatan yang kuat dari penggunanya karena harus mendengarkan seluruh durasi dari *file* video tersebut. Masalah ini dapat diatasi dengan cara memberikan identitas dari setiap *file* video, sehingga ketika didapatkan *file* video yang memiliki identitas yang sama maka *file* video tersebut merupakan *file* video yang duplikat atau ganda. Dalam penelitian ini cara yang digunakan untuk mendapatkan identitas file video tersebut menggunakan fungsi hash SHA-384. Pada bagian analisa ini dipersiapkan *file* video dengan format MP4. File tersebut akan dideteksi keasliannya dengan menggunakan SHA 384 dan akanmenghasilkan kode hash. Pada proses pengambilan nilai *hash* dilakukan terhadap data *binary* dari nilai *byte*. Identitas *file* video yang yang asli dan yang telah dimanipulasi akan dicari nilai final *hash value* dan hasilnya dibandingkan antara keduanya, sehingga ditemukan hasil akhirnya adalah *file* video asli dan *file* video yang telah dimanipulasi memiliki nilai final *hash value* yang tidak sama.

3.2 Penerapan Algoritma SHA-384

Pada bagian analisa ini dipersiapkan *file* video dengan format MP4. *File* tersebut akan dideteksi keasliannya dengan menggunakan *SHA 384* dan akan menghasilkan kode *hash*. Pada proses pengambilan nilai *hash* dilakukan terhadap data

binary dari nilai byte. Identitas file video yang yang asli dan yang telah dimanipulasi akan dicari nilai *final hash value* dan hasilnya dibandingkan antara keduanya, sehingga ditemukan hasil akhirnya adalah file video asli dan file video yang telah dimanipulasi memiliki nilai *final hash value* yang tidak sama. Pada contoh kasus penerapan SHA-384 pada penelitan ini menggunakan objek *Video* dengan spesifikasi, sebagai berikut:

Nama Video : tumbuhan.mp4
Ekstensi/Type : *.mp4
Kapasitas : 1.18 MB (1,241,761 bytes)
Durasi : 00:00:16
Bit rate : 611 kbps

Dalam rangka mempermudah proses analisis, kami telah mengambil sampel dengan ukuran 5 x 5 pixel. Nilai pixel pada sampel tersebut diperoleh melalui penggunaan aplikasi Binary Viewer. Tabel berikut ini memuat nilai-nilai pixel yang telah diambil. Proses pengambilan sampel dan analisis ini menjadi penting dalam penelitian kami untuk memahami karakteristik dan informasi yang terkandung dalam citra yang diuji.

Tabel 1. Nilai Sample Hexadesimal

00	00	00	20	66
69	73	6F	6D	69
00	00	48	9D	6D
00	00	00	00	00
00	00	3E	AD	00

Langkah-langkah penerapan algoritma SHA-384 dalam mendeteksi keaslian file video melibatkan pengubahan nilai pixel dari format heksadesimal menjadi bilangan biner. Proses ini menjadi tahap awal yang penting. Selanjutnya, algoritma SHA-384 digunakan untuk menghasilkan fungsi hash unik untuk setiap video. Fungsi hash ini digunakan untuk membandingkan nilai hash video asli dengan video yang telah mengalami manipulasi. Jika terdapat perbedaan antara nilai hash video asli dan video yang dimanipulasi, ini menunjukkan adanya perubahan pada video tersebut.

Tabel 2. Nilai Hexadesimal Dalam Biner

00000000	00000000	00000000	00100000	01100110
01101001	01110011	01101111	01101101	01101001
00000000	00000000	01001000	10011101	01101101
11010110	01110000	00000000	00000000	00000000
00000001	00000000	00111110	10101101	00000000

a. Penambahan Padding Bit

Dari tabel nilai hexadesimal dalam biner diketahui bahwa panjang X=200 bit. Proses berikutnya adalah dengan menambahkan *padding* bit 1 dan sisanya 0 sejumlah *k*, dengan persamaan sebagai berikut :

$l + 1 + k = 896 \bmod 1024$

l ➡ 5 x 5 = 25
25 x 8 = 200

Jadi, $l + 1 + k = 896 \bmod 1024$ ➡ $200 + 1 + k = 896 \bmod 1024$
$k = 896 - 201 \bmod 1024$
$k = 695 \bmod 1024$
$k = 695$

695

M = 00000000 00000000 00000000 00100000 1000000….0000

b. Penambahan panjang pesan

695 64

M = 00000000 00000000 00000000 00100000 1000000….0000….11001000

Lakukan penambahan bit sebanyak 695 dan penambahan panjang pesan sebanyak 64 bit

Tabel 3. Penambahan Bit dan Penambahan Pesan

$M^{(0)}$	00000000	00000000	00000000	00100000
	01100110	01101001	01110011	01101111
$M^{(1)}$	01101101	01101001	00000000	00000000
	01001000	10011101	01101101	11010110
$M^{(2)}$	01110000	00000000	00000000	00000000
	00000001	00000000	00111110	10101101
$M^{(3)}$	00000000	10000000	00000000	00000000

	00000000	00000000	00000000	00000000
$M^{(4)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(5)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(6)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(7)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(8)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(9)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(10)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(11)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(12)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(13)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(14)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
$M^{(15)}$	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	11001000

c. Penjadwalan pesan

$Mt0 \leq t \leq 15$

$S_1^{(384)}(Wt-2) + Wt-7 + S_0^{(384)}(Wt-15) + Wt-16 \quad 16 \leq t \leq 79$

$S_0^{(384)}(x) = ROTR1(x) \oplus ROTR8(x) \oplus SHR7(x)$

$S_1^{(384)}(x) = ROTR19(x) \oplus ROTR61(x) \oplus SHR6(x)$

$Wt = pesan\ ke\ t\ yang\ baru\ mulai\ dari\ 16\ sampai\ 79$

$W16 = S_1^{(384)}(W16-2) + W16-7 + S_0^{(384)}(W16-15) + W16-16$

$W16 = S_1^{(384)}(W14) + W9 + S_0^{(384)}(W1) + W0$

$S_0^{(384)}(W1) = ROTR1(W1) \oplus ROTR8(W1) \oplus SHR7(W1)$

11110111111101111111011111111001111110011111100111111010111111101 0
11110111111101111111011111111001111110011111100111111010111111101 00000000
11110111111101111111011111111001111110011111100111111010111111101 0000000000

$\oplus$

11111101 11111101 11111110 01111110 01111110 01111110 11111110 10000000

$S_1^{(384)}(W_{14}) = ROTR^{19}(W_{14}) \oplus ROTR^{61}(W_{14}) \oplus SHR^{6}(W_{14})$

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

$\oplus$

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

W_{16} = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
11111101 11111101 11111110 01111110 01111110 01111110 11111110 10000000
11110111111110001111101011111010111110111111101011111010111111011

+

11110101111101101111100101111001011110100111100111111001011101100

$W17 = S_1^{(384)}(W17-2) + W17-7 + S_0^{(384)}(W17-15) + W17-16$

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 2, No 2, February 2023
Hal 81-88
https://ejurnal.seminar-id.com/index.php/bulletinds

$$W17 = S_1^{(384)}(W15) + W10 + S_0^{(384)}(W2) + W1$$
$$S_0^{(384)}(W2) = ROTR1(W2) \oplus ROTR8(W2) \oplus SHR7(W2)$$

11111011111110011111100111111001111110111111101111111001111111000
11111100111111001111110011111101111111011111110011111100 00000000
11111110011111100111111001111110111111101111111001111110000000000
⊕
11111001 01111011011110110111101011111000111110010111011111110000

Diatas adalah langkah penjadwalan pesan yang dilakukan sebanyak 10 putaran. Untuk hasil keseluruhan dari penjadwalan pesan tersebut dibuat kedalam tabel,seperti tabel dibawah berikut:

Tabel 4. Hasil Nilai Penjadwalan Pesan dari Biner ke Hexadesimal

W0=f7f8fafafbfafafb	W21=bb4cb53eeb7ea000	W42=e1f8f118d61d780	W63=f5f6f9797a02f97b
W1=fbfbfbfcfcfcfdfd	W22=c1be0a3a45bbc0c8	W43=efad621e771fc8a0	W64=f5f6f9797a79f97b
W2=fdfcfcfcfdfdfcfc	W23=70d10dc95a21f97b	W44=1f8f118d61d780c8	W65=1536f9797a79f97b
W3=fd80000000000000	W24=33c81ac764e6ddf5	W45=d8054b1e14ad2202	W66=12ef35dc271f4c48
W4=0000000000000000	W25=c97c2de7974aba7c	W46=40d8054b1e14ad00	W67=12ef35dc271f4780
W5=0000000000000000	W26=c81fd33a28227a80	W47=866d46dd1b754e00	W68=12ef35dc271f52c8
W6=0000000000000000	W27=b908026636458e00	W48=6d46dd1b754e0000	W69=10edb5dc271f40c8
W7=0000000000000000	W28=597244d8c81d4000	W49=ad621e771fc8a000	W70=12eff5fc271f40c8
W8=0000000000000000	W29=12ef35dc271f40c8	W50=cfab5138a9785175	W71=96af35dc271f46c8
W9=0000000000000000	W30=eb2884a16172a70b	W51=12ef35dc271f40d0	W72=f2ef35dc271f47c8
W10=0000000000000000	W31=2f5ca1f0d86d4534	W52=52ef35dc271f4040	W73=32ef35dc271f0000
W11=0000000000000000	W32=74bf6e1c4c49cf21	W53=33c81ac764e6dd80	W74=cf8b513829785100
W12=0000000000000000	W33=cf8b513829785175	W54=33c81ac764e6ddc8	W75=cf8b513829780000
W13=0000000000000000	W34=866d46dd1b754ec0	W55=a333c81ac764e6dd	W76=5b3f59784e7ae483
W14=0000000000000000	W35=11eb172ed451f880	W56=74bf6e1c4c49cfa9	W77=5b3f59784e7ae48c
W15=00000000000000c8	W36=65efad621e771fc8	W57=54bb6e5c4c49cf21	W78=5b3f59784e7ae480
W16=f5f6f9797a79f97b	W37=318622954a9a6d1b	W58=74bf6e1c4c49cf00	W79=1f8f118d61d780c8
W17=f5777777fc36abf5	W38=5a3f8dd006f72c72	W59=74bf6e1c4c49c021	
W18=8f728a8e53a45bbc	W39=463afebfd12dc3c6	W60=74a16e1c4c49cf21	
W19=43e23c4a5202fd40	W40=40d8054b1e14ad22	W61=f5f6f9797a790000	
W20=b333c81ac764e6dd	W41=162974cce6f10202	W62=15d6f9797a79f97b	

d. *Initialhash value*
Menginisialisasi 8 variabel kerja a, b, c, d, e, f, g dan h

Tabel 5. Menginisialisasi

a=cbbb9d5dc1059ed8	b=629a292a367cd507	c=9159015a3070dd17	d=152fecd8f70e5939
e=67332667ffc00b31	f=8eb44a8768581511	g=db0c2e0d64f98fa7	h=47b5481dbefa4fa4

$H0(0) = cbbb9d5dc1059ed8$
$H1(0) = 629a292a367cd507$
$H2(0) = 9159015a3070dd17$
$H3(0) = 152fecd8f70e5939$
$H4(0) = 67332667ffc00b31$
$H5(0) = 8eb44a8768581511$
$H6(0) = db0c2e0d64f98fa7$
$H7(0) = 47b5481dbefa4fa4$

e. Mengkompresi pesan

$$T1 = h + \sum_1^{(384)}(e) + Ch(e,f,g) + K_t^{(384)} + Wt$$
$$T2 = \sum_0^{(384)}(a) + Maj(a,b,c)$$
$h = g\ e = d + T1\ b = a$
$g = f\ f = c\ a = T1 + T2$
$f = e\ g = b$

keterangan:
$Ch(x,y,z) = (x \wedge y) \oplus (x \wedge z)$
$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$

$$\sum_{0}^{(384)} (x) = ROTR28(x) \oplus ROTR34(x) \oplus ROTR39(x)$$

$$\sum_{1}^{(384)} (x) = ROTR14(x) \oplus ROTR18(x) \oplus ROTR41(x)$$

Setelah menginisialisasi 8 variabel tersebut, selanjutnya hasil akhir dari perhitungan dijumlahkan dengan nilai *value.*

$H0^{(1)}$ = cbbb9d5dc1059ed8 +14e986342ddced0f = e0a52391eee28be7
$H1^{(1)}$ = 629a292a367cd507 +abab4b0ca75a17c7 = e457436ddd6ecce
$H2^{(1)}$ = 9159015a3070dd17 +4abe0af6a67db2fe = dc170c50d6ee9015
$H3^{(1)}$ = 152fecd8f70e5939 + 5cdf6c58fc052572 = 720f5931f3137eab
$H4^{(1)}$ = 67332667ffc00b31 + 8347f5736531b3ec = ea7b1bdb64f1bf1d
$H5^{(1)}$ = 8eb44a8768581511 + d42c3a57cfa78513 =62e084df37ff9a24
$H6^{(1)}$ =db0c2e0d64f98fa7 + 500f7b61186f6c2e =2b1ba96e7d68fbd5
$H7^{(1)}$ = 47b5481dbefa4fa4 + 4abe0af6a67db2fe = 92735314657802a2

Tabel 6. Hasil Nilai SHA-384

e0a52391eee28be7
e457436ddd6ecce
dc170c50d6ee9015
720f5931f3137eab
ea7b1bdb64f1bf1d
62e084df37ff9a24
2b1ba96e7d68fbd5
92735314657802a2

Berdasarkan dari perhitungan sampel diatas diperoleh nilai SHA-384 berbentuk bilangan hexadesimal 8 karakter 128 byte ,yaitu
”e0a52391eee28be7e457436ddd6eccedc170c50d6ee9015720f5931f3137eabea7b1bdb64f1bf1d62e084df37ff9a242b1ba96e7d68fbd592735314657802a2”.

4. KESIMPULAN

Dalam kesimpulan penelitian ini, dapat disimpulkan bahwa Algoritma SHA-384 merupakan alat yang sangat efektif dalam pendeteksian perubahan pada video. Cara kerja algoritma ini menghasilkan fungsi hash yang unik untuk setiap video, yang memungkinkan identifikasi perubahan dengan tingkat akurasi yang tinggi. Penelitian ini dengan jelas menunjukkan bahwa Algoritma SHA-384 memiliki kemampuan untuk mendeteksi perubahan sekecil apa pun pada video awal dan video yang telah dimanipulasi. Perbedaan nilai hash yang dihasilkan oleh video asli dan video manipulasi memberikan bukti konkret tentang kemampuan algoritma ini untuk memonitor dan mendeteksi perubahan dengan presisi yang tinggi. Selanjutnya, penelitian ini juga mengungkapkan bahwa nilai hash video asli dan video manipulasi selalu berbeda. Dalam proses perbandingan nilai hash, tidak ada kesamaan antara keduanya. Ini menunjukkan bahwa Algoritma SHA-384 berhasil dalam mendeteksi perubahan pada video dengan metode membandingkan nilai *hash* video yang asli dengan yang telah dimanipulasi. Dengan kata lain, algoritma ini memiliki kemampuan untuk membedakan antara video asli dan video manipulasi. Kesimpulan ini menggambarkan bahwa Algoritma SHA-384 adalah alat yang sangat andal untuk menjaga integritas video dan dapat digunakan untuk mendeteksi tindakan manipulasi video dengan tingkat akurasi yang sangat tinggi. Implikasinya, algoritma ini memiliki berbagai aplikasi yang luas dalam bidang keamanan video, forensik digital, dan banyak konteks lain yang memerlukan verifikasi dan pengamanan data video. Dengan demikian, penelitian ini telah membuktikan bahwa Algoritma SHA-384 adalah alat yang penting dan efektif dalam mengatasi masalah deteksi perubahan pada video, dan hasilnya dapat memberikan kontribusi yang signifikan dalam berbagai disiplin ilmu yang bergantung pada integritas dan validitas data video.

REFERENCES

[1] D. R. F. Sudarto, “Sistem Deteksi Pemalsuan Video Menggunakan Analisis Forensic Digital,” 2022.
[2] P. M. Simanullang, S. Sinurat, and I. Saputra, “ANALISA METODE SHA384 UNTUK MENDETEKSI ORISINALITAS CITRA DIGITAL,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, 2019.
[3] H. Pasaribu and M. F. Harahap, “PERANCANGAN APLIKASI HASHING FILE MENGGUNAKAN SHA–224 BERBASIS ANDROID,” *J. Mantik Penusa*, vol. 4, no. 1, pp. 19–26, 2020.
[4] M. Ipdal, “Analisa Metode SHA-512 Untuk Tanda Tangan Digital Pada File Video,” *J. Informatics Manag. Inf. Technol.*, vol. 1, no. 1, pp. 23–29, 2021.
[5] L. Silalahi and A. Sindar, “Penerapan Kriptografi Keamanan Data Administrasi Kependudukan Desa Pagar Jati Menggunakan SHA-1,” *J. Nas. Komputasi dan Teknol. Inf*, vol. 3, no. 2, 2020.
[6] M. H. Santoso, N. D. Girsang, H. Siagian, A. Wahyudi, and B. A. Sitorus, “Perbandingan Algoritma Kriptografi Hash MD5 dan

SHA-1," in *Semantika (Seminar Nasional Teknik Informatika)*, 2019, vol. 2, no. 1, pp. 54–59.
[7] B. H. Situmorang, S. Sinurat, and K. Tampubolon, "Implementasi Algoritma Atbash Untuk Menyandikan Pesan Teks Berbasis Android," *Pelita Inform. Inf. dan Inform.*, vol. 7, no. 2, pp. 157–161, 2018.
[8] N. A. O. Saputri, "Implementasi Pengamanan Data Dan Informasi Di Balai Desa Tanding Marga Dengan Metode Steganografi Lsb dan Algoritma Kriptografi Aes," *INFORMANIKA*, vol. 7, no. 01, 2021.
[9] S. G. Damping, "Pengamanan Transmisi Data Iot Kriptografi Aes Pada Sistem Monitoring Kualitas Air Kolam." Universitas Komputer Indonesia, 2022.
[10] S. Erbeliza, "Analisis Keamanan Aplikasi Mobile Commerce Menggunakan Mobile Security Framework (Mobsf) dan Owasp Mobile Application Security Testing Guide (Mastg)." Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
[11] A. Putra, V. Sihombing, and M. H. Munandar, "Rancang bangun aplikasi deteksi tepi citra digital menggunakan algoritma prewitt," *J. Tekinkom (Teknik Inf. dan Komputer)*, vol. 4, no. 1, pp. 83–87, 2021.
[12] P. Pandu Pratama Putra and D. Dafwen Toresa, "BUKU AJAR KEAMANAN INFORMASI DAN JARINGAN KOMPUTER." LPPM Universitas Lancang Kuning, 2021.
[13] R. Gunawan, R. Malfiany, and H. Y. Pane, "Penerapan Digital Marketing Sebagai Strategi Pemasaran Ukm Rempeyek Nok Uus Dengan Video Cinematic Didukung Motion Grafis," *Pixel J. Ilm. Komput. Graf.*, vol. 14, no. 1, pp. 25–36, 2021.
[14] S. M. Myint, M. M. Myint, and A. A. Cho, "A study of SHA algorithm in cryptography," *Int. J. Trend Sci. Res. Dev*, vol. 3, pp. 1453–1454, 2019.
[15] I. B. Senkyire and Q.-A. Kester, "A Cryptographic Tamper Detection Approach for Storage and Preservation of Forensic Digital Data Based on SHA 384 Hash Function," in *2021 International Conference on Computing, Computational Modelling and Applications (ICCMA)*, 2021, pp. 159–164.