RESOLUSI : Rekayasa Teknik Informatika dan Informasi
ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

Implementasi Algoritma SHA-256 Pada Aplikasi Duplicate Document Scanner

Sopiana Nainggolan

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: sopiananainggolan27@gmail.com

Abstrak-Di dalam komputer apabila kita menggandakan sebuah file maka file hasil penggandaan tersebut tidak ada bedanya dengan file yang pertama dan hal ini dapat dilakukan dengan waktu yang cukup singkat. Dalam hal ini yang akan di gandakan yaitu file dokumen word berekstensi docx. SHA(Secure hash Algorithm) adalah salah satu algoritma hash yang relatif masih baru. Pemilihan algoritma SHA-256 ini tidak terlepas dari beberapa penelitian terdahulu yang telah meneliti kinerja algoritma ini. Algoritma SHA-256 mampu mendeteksi perubahan yang terjadi pada citra digital walaupun perubahan yang terjadi hanya 1 piksel saja, perbedaan kecil yang terjadi pada citra tersebut menghasilkan perbedaan yang sangat signifikan dari nilai hash-nya dengan tingkat akurasi mencapai 100%.SHA – 256 menghasilkan message digest dengan panjang 256 bit. SHA – 256 tergolong aman karena didesain sedemikian rupa sehingga tidak memungkinkan mendapatkan pesan yang berhubungan dengan message digest yang sama.

Kata Kunci: Kriptografi; Duplicate; Document Scanner; SHA-256

Abstract-On a computer, if we duplicate a file, the duplicated file is no different from the first file and this can be done in a fairly short time. In this case, what will be duplicated is a word document file with docx extension. SHA (Secure hash Algorithm) is a hash algorithm that is relatively new. The selection of the SHA-256 algorithm is inseparable from several previous studies that have examined the performance of this algorithm. The SHA-256 algorithm is able to detect changes that occur in digital images even though the changes that occur are only 1 pixel, the small differences that occur in the image produce a very significant difference from the hash value with an accuracy level of up to 100%. SHA - 256 produces a message digest with 256 bits in length. SHA - 256 is classified as safe because it is designed in such a way that it is not possible to get messages related to the same message digest.

Keywords: Cryptography; Duplicate; Document Scanner; SHA-256

1. PENDAHULUAN

Manusia telah melewati berbagai macam revolusi dibidang industri, dan saat ini manusia berada di era revolusi industri 4.0. Pada era revolusi industri 4.0 ini komputer berkembang sangat pesat, memiliki kemampuan yang sangat cepat, tetapi masih terdapat beberapa masalah yang belum dapat diselesaikan dan perkembangannya tergolong lambat, yaitu perkembangan media penyimpanan atau *storage*. Untuk itu masih diperlukan manajemen *file* yang dapat mengoptimalkan penggunaan ruang penyimpanan.

Didalam komputer apabila kita menggandakan sebuah *file* maka *file* hasil penggandaan tersebut tidak ada bedanya dengan *file* yang pertama dan hal ini dapat dilakukan dengan waktu yang cukup singkat. Dalam hal ini yang akan di gandakan yaitu *file* dokumen word berekstensi docx.

Terkadang secara tidak sengaja seseorang menyimpan *file* docx yang sama didalam sebuah media penyimpanan, tentu hal ini akan menyita ruang penyimpanan yang bersifat terbatas tadi. Dan tentunya ini sangat tidak efektif karena menyimpan *file* docx yang sama lebih dari satu. Untuk itu seseorang perlu melakukan manajemen penyimpanan *file* agar ruang penyimpanan dapat dimanfaatkan secara optimal. Tantangan yang dihadapi ketika seseorang ingin memanajemen *file* dalam ruang penyimpanan tersebut adalah membedakan *file* yang satu dengan *file* yang lain. Sudah barang tentu cara yang dilakukan adalah melihat isi *file* tersebut satu persatu. Hal ini sangat tidak efektif dan memakan waktu yang cukup lama, terlebih jika *file* yang akan di periksa tersebut jumlahnya banyak.

Fungsi *hash* dalam kriptografi memiliki varian yang sangat banyak dan memiliki kelebihan dan kelemahan masing-masing. Tingkat keefektifan dari algoritma yang digunakan tergantung pada kasus apa yang akan diselesaikan. Semakin sulit dan panjang proses yang dilalui untuk menghasilkan sebuah nilai *hash* maka akan memakan waktu yang lama pula, tetapi tingkat keamanannya juga tinggi. Didalam *Secure Hash Algorithm*(SHA) terdapat beberapa varian yaitu SHA-1, SHA-256, SHA-384, dan SHA-512. Angka 256, 385 dan 512 adalah panjang nilai *hash* yang dihasilkan. Dalam penelitian ini penulis menggunakan SHA-256 yang akan digunakan untuk mencari sidik jari atau identitas dari *file* docx sehingga *file-file* docx yang ganda dapat di hapus salah satu dan akhirnya akan mengoptimalkan penggunaan ruang penyimpanan. Algoritma SHA-256 juga dapat digunakan untuk melakukan pengecekan integritas data, pembuatan *digital signature*, dan lain-lain. Terdapat juga fungsi *hash* pada jurnal yang berjudul Analisis kecepatan dan Keamanan algoritma *Secure Hash Algorithm* 256 (SHA-256) untuk otentikasi pada pesan teks. Fungsi otentikasi pesan yang biasa digunakan adalah fungsi *hash*,dengan menggunakan fungsi *hash* satu arah maka akan dihasilkan *message digest* dari pesan asli [1]. Metode SHA-256 ini terdapat juga pada jurnal Implementasi Aplikasi Digital Signature menggunakan fungsi *hash*. Algoritma SHA-256 dan RSA di Badan Nasional kota Cimahi. Hal ini digunakan sebagai mekanisme dan teknik untuk melindungi suatu yang dapat berupa data atau informasi di dalam sistem [2].

Pemilihan algoritma SHA-256 ini tidak terlepas dari beberapa penelitian terdahulu yang telah meneliti kinerja algoritma ini. Algoritma SHA-256 mampu mendeteksi perubahan yang terjadi pada citra digital walaupun perubahan

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

yang terjadi hanya 1 piksel saja, perbedaan kecil yang terjadi pada citra tersebut menghasilkan perbedaan yang sangat signifikan dari nilai *hash*-nya dengan tingkat akurasi mencapai 100% [3].

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Secara etimologi kata kriptografi (*Cryptography*) berasal dari bahasa Yunani, yaitu kryptos yang artinya yang tersembunyi dan graphein yang artinya tulisan. awal mula kriptografi dipahami sebagai ilmu tentang menyembunyikan pesan , tetapi seiring perkembangan zaman hingga saat ini pengertian kriptografi berkembang menjadi ilmu tentang teknik matematis yang digunakan untuk menyelesaikan persoalan keamanan berupa privasi dan otentikasi [5].

2.2 Algoritma SHA-256

SHA*(SecurehashAlgorithm)* adalah salah satu algoritma *hash* yang relatif masih baru. Algoritma ini dirancang oleh *The National Institute of Standards and Technology* (NIST) pada tahun 2002. SHA – 256 menghasilkan *message digest* dengan panjang 256 *bits*. SHA – 256 tergolong aman karena didesain sedemikian rupa sehingga tidak memungkinkan mendapatkan pesan yang berhubungan dengan *message digest* yang sama. Seperti pada contoh kasus dokumen citra digital hasil pemindaian dari ijazah dan transkrip nilai. Proses untuk menghasilkan *message digest* pada algoritma ini meliputi lima tahapan.

1. *MessagePadding*
   Input pesan pada algoritma SHA-256 akan dibagi menjadi blok-blok yang masing-masing panjangnya adalah 512 bit. Akibat dari pembagian ini maka jumlah blok terakhir akan lebih kecil atau sama dengan 512 bit. Blok terakhir tersebut akan mengalami *message padding.* Langkah-langkah *message padding* adalah sebagai berikut:
   a. Diawali dengan masuknya input pesan yang memiliki kode *American Standard Code for Information Interchange* (ASCII) dan kemudian diubah ke dalam bentuk biner rangkaian bit yang akan dihitung panjangnya.
   b. Rangkaian bit tersebut dibagi menjadi blok-blok yang masing-masing mempunyai panjang 512 bit. Hasil pembagian akan menyebabkan jumlah blok terakhir lebih kecil satu atau sama dengan 512 bit.
   c. Lakukan penambahan bit-bit isian *(padding)* pada blok terakhir pesan tersebut. Bit-bit yang digunakan sebagai bit isian adalah bit ‘1’ diikuti sejumlah bit ‘0’ sesuai dengan kebutuhan, dengan ketentuan sebagai berikut:
      1. Jika panjang bit blok pesan terakhir lebih kecil dari 448 bit, maka ditambahkan bit ‘1’ pada posisi bit paling akhir, diikuti dengan beberapa bit ‘0’ sedemikian sehingga total panjang bit setelah proses tersebut adalah adalah 448 bit.
      2. Jika panjang bit blok pesan terakhir lebih besar atau sama dengan 448 bit, maka ditambahkan bit ‘1’ pada posisi bit paling terakhir, diikuti dengan beberapa bit ‘0’ sedemikian sehingga total panjang bit setelah proses tersebut 512 bit. Kemudian dibuat 448 bit baru yang isi bitnya ‘0’.
   d. Jika panjang bit blok pesan terakhir sama dengan 512 bit, maka harus dibuat blok baru untuk menampung proses *message padding*. Bit pertama dari blok baru diisi bit’1’, sedangkan bit-bit berikutnya sampai dengan panjang bit 448 diisi oleh bit ‘0’. Jumlah total bit isian yang ditambahkan adalah 448 bit.
2. Penambahan Panjang Bit
   Setelah proses *messagepadding*, jumlah bit pada blok terakhir adalah 448 bit. Representasikan M ke dalam bilangan biner untuk memperoleh 64 bit terakhirnya agar total panjang blok terakhir 512 bit.
   a. Urutan byte paling kanan dari nilai representasi panjang pesan (M) dijadikan *low order*.
   b. Tambahkan representasi M tersebut pada 448 bit terakhir, sehingga jumlah panjang blok terakhir adalah 512 bit.
3. Inisialisasi Nilai *Hash*Awal
   Pada SHA – 256 untuk menyimpan nilai inisialisasi awal dan nilai output sementara digunakan *buffer* H0, H1, H2, H3, H4, H5, H6, H7. Di sisi lain, untuk penyimpanan proses sementatat digunakan *buffer* a, b, c, d, e, f, g, h. Nilai H0, H1, H2, H3, H4, H5, H6, H7 untuk inisialisasi awal dalam notasi heksadesimal.

Table 1. Inisialisasi awal dalam notasi heksadesimal

H0	6a09e667
H1	bb67ae85
H2	3c6ef372
H3	a54ff53a
H4	510e527f
H5	9b0588c
H6	1f83d9ab
H7	5be0cd19

4. Pemrosesan
   Pemrosesan merupakan bagian inti yang terdiri atas 1 *round* mempunyai 64 operasi. Untuk memproses setiap satu blok pesan 512 bit diperlukan 64 operasi. Setiap blok pesan M(1), M(2), ... , M(n) dengan N adalah jumlah blok pesan. Untuk setiap blok pesan akan dilakukan langkah-langkah:
   a. Persiapkan penjadwalan pesan.

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

b. Inisialisasi*working* variabel a, b, c, d, e, f, g dan h, untuk M(1) dengan nilai *hash* awal
   a = H0(I-1)
   b = H1(i-1)
   c = H2(i-1)
   d = H3(i-1)
   e = H4(i-1)
   f = H5(i-1)
   g = H6(i-1)
   h = H7(i-1)
c. Hitung masing-masing penjadwalan.
d. Menghitung nilai *hash* perantara untuk masing-masing blok pesan.

5. *Output*
   Output diperoleh setelah semua blok M(N) 512 bit diproses. Setelah semua langkah pemprosesan dilakukan sejumlah N kali, maka akan didapat 256 bit *message digest*.

3. HASIL DAN PEMBAHASAN

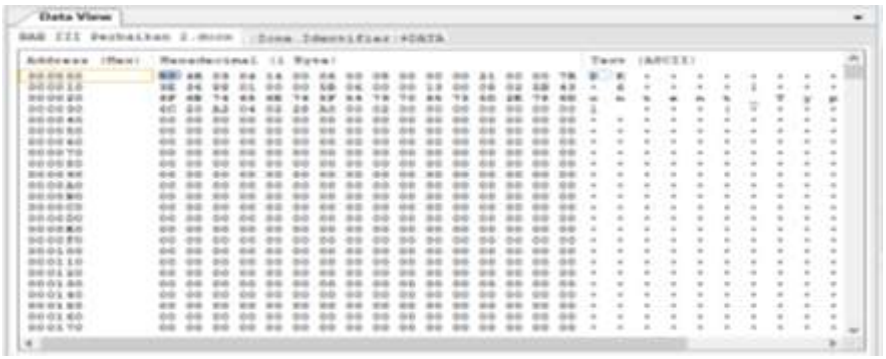
3.1 Analisa Masalah

Ketika menghadapi masalah dalam melakukan pencarian *file* dokumen yang duplikat atau ganda pada sebuah media penyimpanan membutuhkan ketelitian dan ingatan yang kuat dari penggunanya karena harus melihat isi dokumen satu persatu. Masalah ini dapat diatasi dengan cara memberikan identitas dari setiap *file* dokumen, sehingga ketika didapatkan *file* dokumen yang memiliki identitas yang sama maka *file* dokumen tersebut merupakan *file* dokumen yang duplikat atau ganda. Dalam penelitian ini cara yang digunakan untuk mendapatkan identitas *file* dokumen tersebut menggunakan Algoritma SHA-256.

Langkah awal yang dilakukan ketika ingin melakukan pencarian *file* dokumen yang ganda atau duplikat adalah melakukan *scanning* pada sebuah ruang penyimpanan yang di dalamnya terdapat banyak *file* dokumen. Setelah melakukan proses *scanning* maka langkah selanjutnya adalah membangkitkan identitas dari *file* dokumen hasil *scanning* tersebut. Setelah identitas dari *file* citra tersebut berhasil di dapatkan maka langkah selanjutnya adalah melakukan pengelompokkan *file* dokumen berdasarkan identitas *file* dokumen yang di bangkitkan menggunakan fungsi *hash* SHA-256.

3.1.1 Penerapan Algoritma SHA-256

Objek pada penelitian ini adalah *file* dokumen dengan ekstensi docx berukuran 100 KB. Untuk memudahkan proses analisa maka diambil sampel dari *file* dokumen menggunakan aplikasi Binary Viewer dan diambil sampel sebanyak 25 byte.



Gambar 1. Nilai sampel *file* dokumen

Nilai-nilai dalam bentuk heksadesimal tersebut yang akan diproses dengan menggunakan Algoritma SHA-256 untuk mengetahui identitas dari sebuah *file* dokumen itu sendiri.

Tabel 1. Nilai File Dokumen 25 Byte

Address (heksadesimal)	Value Heksadesimal	Address (heksadesimal)	Value Heksadesimal
00 00 00	50	00 00 0D	00
00 00 01	4B	00 00 0E	00
00 00 02	03	00 00 0F	7B
00 00 03	04	00 00 10	9E
00 00 04	14	00 00 11	36
00 00 05	00	00 00 12	88
00 00 06	06	00 00 13	01

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

Address (heksadesimal)	Value Heksadesimal	Address (heksadesimal)	Value Heksadesimal
00 00 07	00	00 00 14	00
00 00 08	08	00 00 15	00
00 00 09	00	00 00 16	5B
00 00 0A	00	00 00 17	06
00 00 0B	00	00 00 18	00
00 00 0C	21	00 00 19	00

Berikut ini langkah-langkah penerapan metode Algoritma SHA-256 untuk mengetahui duplikat *file* dokumen, dimana sebelumnya terlebih dahulu dilakukan pengubahan nilai heksadesimal dari *file* dokumen menjadi bilangan biner.

Tabel 2. Input Nilai Biner

01010000	01001011	00000011	00000100	00010100
00000000	00000110	00000000	00001000	00000000
00000000	00000000	00100001	00000000	00000000
01111011	10011110	00110110	10001000	00000001
00000000	00000000	01011011	00000110	00000000

1. Penambahan *Padding Bit*
   Padding bit di tambahkan karena algoritma SHA-256 membutuhkan minimal satu blok data *input* dengan panjang 512 bit, sehingga jika *input* kurang dari 512 bit maka di tambahkan *padding bit* yang di mulai dengan 1 selanjutnya di tambahkan 0.
   k =*l*+ 1 = 448 mod 512
   k = 200 + 1 = 448 mod 512
   k = 201= 448 mod 512
   k = 448 – 201
   k = 247
   Maka banyaknya *padding bit* 1 dan selanjutnya nilai 0 sebanyak 247 dapat di lihat pada tabel 3.3. di bawah ini:

Tabel 3. Penambahan *padding* bit

01010000	01001011	00000011	00000100	00010100	00000000	00000110	00000000
00001000	00000000	00000000	00000000	00100001	00000000	00000000	01111011
10011110	00110110	10001000	00000001	00000000	00000000	01011011	00000110
00000000	**10000000**	**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**

2. Penambahan Panjang *Append*
   Penambahan panjang *append* dilakukan dengan penambahan panjang data sebanyak 64 bit di akhir. Panjang data adalah 200 bit sehingga ditambahkan panjang *append* 11001000 di akhir sebanyak 64 bit sebagai berikut:

Tabel 4. Penambahan Panjang *Append*

01010000	01001011	00000011	00000100	00010100	00000000	00000110	00000000
00001000	00000000	00000000	00000000	00100001	00000000	00000000	01111011
10011110	00110110	10001000	00000001	00000000	00000000	01011011	00000110
00000000	10000000	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**	**11001000**

3. *Parsing* Data
   Pada kasus ini panjang data tidak lebih dari 512 sehingga hanya menghasilkan 1 blok 512bit yaitu $M_0^{(0)}$ sampai $M_{15}^{(0)}$. Tahap selanjutnya adalah melakukan *parsing* data dengan membagi setiap blok 512 bit menjadi 16 blok berukuran 32 bit.

Tabel 5.. *Parsing* Data

Data		Biner	Heksadesimal
$M_0^{(0)}$	:	01010000 01001011 00000011 00000100	504B0304
$M_1^{(0)}$	:	00010100 00000000 00000110 00000000	14000600
$M_2^{(0)}$	:	00001000 00000000 00000000 00000000	08000000

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

Data		Biner	Heksadesimal
$M_3^{(0)}$	:	00100001 00000000 00000000 01111011	2100007B
$M_4^{(0)}$	:	10011110 00110110 10001000 00000001	9E368801
$M_5^{(0)}$	:	00000000 00000000 01011011 00000110	00005B06
$M_6^{(0)}$	:	00000000 10000000 00000000 00000000	00800000
$M_7^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_8^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_9^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_{10}^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_{11}^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_{12}^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_{13}^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_{14}^{(0)}$	:	00000000 00000000 00000000 00000000	00000000
$M_{15}^{(0)}$	:	00000000 00000000 00000000 11001000	000000C8

4. Inisialisasi Nilai *Hash*
   Setelah proses *parsing* data maka langkah selanjutnya adalah inisialisasi nilai *hash* di mana nilai ini merupakan sebuah ketentuan yaitu:

Tabel 6. Inisial *hashvalue*

Variabel	HashValue
$H_0^{(0)}$	6A09E667
$H_1^{(0)}$	BB67EA85
$H_2^{(0)}$	3C6EF372
$H_3^{(0)}$	A54FF53A
$H_4^{(0)}$	510E527F
$H_5^{(0)}$	9B05688C
$H_6^{(0)}$	1F83D9AB
$H_7^{(0)}$	5BE0CD19

5. Penjadwalan Data
   Kemudian dilakukan proses penjadwalan data, langkah inidiawali dengan mengubah setiap blok data menjadi bilangan heksadesimal dengan ketentuan sebagai berikut:

$$Wt = \begin{cases} M_t^{(t)} & 0 \le t \le 15 \\ \sigma_1^{(256)}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16}, & 16 \le t \le 63 \end{cases}$$

Tabel 7. Penjadwalan Data

504B0304	14000600	08000000	2100007B	9E368801	00005B06	00800000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	000000C8
51CB0804	157D0600	F136D02D	388C28CB	7C579322	0016A53E	3C6A650D	D803CDA4
93A52921	51C7C81F	06DC3BB7	996FC64B	9AC379F9	20754978	945B1CA3	61DC16B1
F8B8DE17	4156E53F	B130175F	31328253	68C2B93D	2A984B71	74EE8AC4	F8B44D67
2FC0408A	B4150C93	F5FC2130	B87CB8CF	D6AF2C53	3C7F3916	84BA06DF	F4A6B209
390AA82D	F49F8263	26066D1D	AFE8DA47	96411C09	5D983C05	DFB3C597	4CD95C64
1CBD2259	F6E1546B	2191FE6F	646F614A	DAEC0F2C	46F23A07	8D189677	25CE0270

Untuk menjadwalkan datake 16 sampai 63 dilakukan perhitungan sebagai berikut:

Data ke 16

$\sigma_1^{(256)}(W_{i-2}) = ((W_{i-2})ROTR\ 17) \oplus ((W_{i-2})ROTR\ 19) \oplus ((W_{i-2})SHR10)$

$((W_{16-2})ROTR\ 17)$ $=((W_{14})ROTR\ 17)$
$= ((00000000)ROTR\ 17)$
$=(00000000)$

$((W_{16-2})ROTR\ 19)$ $=((W_{14})ROTR\ 19)$
$= ((00000000)ROTR\ 19)$
$=(00000000)$

$((W_{16-2})SHR10)$ $=((W_{14})SHR10)$
$=((00000000)SHR10)$
$= (00000000)$

$\sigma_1^{(256)}(W_{i-2}) = (00000000) \oplus (00000000) \oplus (00000000)$
$=(\mathbf{00000000})$

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

W_{16-7} = W_9
= **00000000**
$\sigma_0^{(256)}(W_{i-15}) = ((W_{i-15})ROTR\ 7) \oplus ((W_{i-15})ROTR\ 18) \oplus ((W_{i-15})SHR3)$
$((W_{16-15})ROTR\ 7)$ =$((W_1)ROTR\ 7)$
= $((14000600)ROTR\ 7)$
= $(0028000C)$
$((W_{16-15})ROTR\ 18)$ =$((W_1)ROTR\ 18)$
= $((14000600)ROTR\ 18)$
=(01800500)
$((W_{16-15})SHR\ 3)$ =$((W_1)SHR\ 3)$
= $((14000600)SHR\ 3)$
=$(028000C0)$
$\sigma_0^{(256)}(W_{i-15}) = (028000C0) \oplus (01800500) \oplus (028000C0)$
=(**01800500**)
W_{16-16} = W_0
= **504B0304**
$Wt = \sigma_1^{(256)}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16}$
$Wt = 00000000 + 00000000 + 01800500 + 504B0304$
$Wt =$ **51*CB*0804**

Data ke 17
$\sigma_1^{(256)}(W_{i-2}) = ((W_{i-2})ROTR\ 17) \oplus ((W_{i-2})ROTR\ 19) \oplus ((W_{i-2})SHR10)$
$((W_{17-2})ROTR\ 17)$ =$((000000C8)ROTR\ 17)$
= $((00000000)ROTR\ 17)$
=(00640000)
$((W_{17-2})ROTR\ 19)$ =$((W_{15})ROTR\ 19)$
= $((000000C8)ROTR\ 19)$
=(00190000)
$((W_{17-2})SHR10)$ =$((W_{15})SHR10)$
=$((000000C8)SHR10)$
= (00000000)
$\sigma_1^{(256)}(W_{i-2}) = (00640000) \oplus (00190000) \oplus (00000000)$
=(**007*D*0000**)
W_{17-7} = W_{10}
= **00000000**
$\sigma_0^{(256)}(W_{i-15}) = ((W_{i-15})ROTR\ 7) \oplus ((W_{i-15})ROTR\ 18) \oplus ((W_{i-15})SHR3)$
$((W_{17-15})ROTR\ 7)$ =$((W_2)ROTR\ 7)$
= $((08000000)ROTR\ 7)$
= (00100000)
$((W_{17-15})ROTR\ 18)$ =$((W_2)ROTR\ 18)$
= $((08000000)ROTR\ 18)$
=(00000200)
$((W_{17-15})SHR\ 3)$ =$((W_2)SHR\ 3)$
= $((08000000)SHR\ 3)$
=(01000000)
$\sigma_0^{(256)}(W_{i-15}) = (00100000) \oplus (00000200) \oplus (01000000)$
=(**01100200**)
W_{17-16} = W_1
= **14000600**
$Wt = \sigma_1^{(256)}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16}$
$Wt = 007D0000 + 00000000 + 01000000 + 14000600$
$Wt =$ **157*D*0600**

Data ke 18
$\sigma_1^{(256)}(W_{i-2}) = ((W_{i-2})ROTR\ 17) \oplus ((W_{i-2})ROTR\ 19) \oplus ((W_{i-2})SHR10)$
$((W_{18-2})ROTR\ 17)$ =$((W_{16})ROTR\ 17)$

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

$= ((51CB0804)ROTR\ 17)$
$=(840228E5)$

$((W_{18-2})ROTR\ 19)$ $=((W_{16})ROTR\ 19)$
$= ((51CB0804)ROTR\ 19)$
$=(61008A39)$

$((W_{18-2})SHR10)$ $=((W_{16})SHR10)$
$=((51CB0804)SHR10)$
$= (001472C2)$

$\sigma_1^{(256)}(W_{i-2}) = (840228E5) \oplus (61008A39) \oplus (001472C2)$
$=(\boldsymbol{E516D01E})$

W_{18-7} $= W_{11}$
$=$ **00000000**

$\sigma_0^{(256)}(W_{i-15}) = ((W_{i-15})ROTR\ 7) \oplus ((W_{i-15})ROTR\ 18) \oplus ((W_{i-15})SHR3)$

$((W_{18-15})ROTR\ 7)$ $=((W_3)ROTR\ 7)$
$= ((2100007B)ROTR\ 7)$
$= (F6420000)$

$((W_{18-15})ROTR\ 18)$ $=((W_3)ROTR\ 18)$
$= ((2100007B)ROTR\ 18)$
$=(001EC840)$

$((W_{18-15})SHR\ 3)$ $=((W_3)SHR\ 3)$
$= ((2100007B)SHR\ 3)$
$=(0420000F)$

$\sigma_0^{(256)}(W_{i-15}) = (F6420000) \oplus (001EC840) \oplus (0420000F)$
$=(\boldsymbol{F27CC84F})$

W_{18-16} $= W_2$
$=$ **08000000**

$Wt = \sigma_1^{(256)}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16}$
$Wt = E516D01E + 00000000 + 0420000F + 08000000$
$Wt = \boldsymbol{F136D02D}$

6. InisialisasiVariabel kerja
Selanjutnya melakukan inisialisasi variabel kerja *a*, *b*, *c*, *d*, *e*, *f*, *g* dan *h* di mana setiap variabel diambil dari *initialhashvalue a=H0(0), b=H1(0), c=H2(0) , d=H3(0), e=H4(0), f=H5(0), g=H6(0), h=H7(0)*. Selanjutnya dilakukan proses komputasi fungsi *hash*SHA-256 dari t=0 sampai t=63.

Tabel 8. Inisialisasi Variabel Kerja

	a	***b***	***c***	***d***	***e***	***f***	***g***	***h***
Init	6A09E667	BB67AE85	3C6EF372	A54FF53A	510E527F	9B05688C	1F83D9AB	5BE0CD19
t=0	1C538B51	6A09E667	BB67AE85	3C6EF372	E912E5A6	510E527F	9B05688C	1F83D9AB
t=1	1BE9E4FC	1C538B51	6A09E667	BB67AE85	BA4B9F78	E912E5A6	510E527F	9B05688C
t=2	0AD24169	1BE9E4FC	1C538B51	6A09E667	2566B35A	BA4B9F78	E912E5A6	510E527F
t=3	106CC600	0AD24169	1BE9E4FC	1C538B51	5EA9880B	2566B35A	BA4B9F78	E912E5A6
t=4	7FC4B549	106CC600	0AD24169	1BE9E4FC	FA2BC98B	5EA9880B	2566B35A	BA4B9F78
t=5	68C2A8C6	7FC4B549	106CC600	0AD24169	9B07DB68	FA2BC98B	5EA9880B	2566B35A
t=6	1980FE34	68C2A8C6	7FC4B549	106CC600	EE370C4D	9B07DB68	FA2BC98B	5EA9880B
t=7	A34FCCF1	1980FE34	68C2A8C6	7FC4B549	45C3B2C1	EE370C4D	9B07DB68	FA2BC98B
t=8	9D8EC37A	A34FCCF1	1980FE34	068C2A8C	ECE648F4	45C3B2C1	EE370C4D	9B07DB68
t=9	D3B2401C	9D8EC37A	A34FCCF1	1980FE34	B8F4E55A	ECE648F4	45C3B2C1	EE370C4D
t=10	E69D3D38	D3B2401C	9D8EC37A	A34FCCF1	F0873C29	B8F4E55A	ECE648F4	45C3B2C1
t=11	354987C4	E69D3D38	D3B2401C	9D8EC37A	5C667B40	F0873C29	B8F4E55A	ECE648F4
t=12	B788F812	354987C4	E69D3D38	D3B2401C	4811DB08	5C667B40	F0873C29	B8F4E55A
t=13	81405D99	B788F812	354987C4	E69D3D38	4ED11808	4811DB08	5C667B40	F0873C29
t=14	708CB16D	81405D99	B788F812	354987C4	13F695B4	4ED11808	4811DB08	5C667B40
t=15	9D715D26	708CB16D	81405D99	B788F812	3BA4CDD5	13F695B4	4ED11808	4811DB08
t=16	2EF9737F	9D715D26	708CB16D	81405D99	A3DD4CC9	3BA4CDD5	13F695B4	4ED11808
t=17	7A88E68F	2EF9737F	9D715D26	708CB16D	F79A05ED	A3DD4CC9	3BA4CDD5	13F695B4
t=18	904B8569	7A88E68F	2EF9737F	9D715D26	F89AEF99	F79A05ED	A3DD4CC9	3BA4CDD5
t=19	01DCFF0A	0904B856	7A88E68F	2EF9737F	0338AF7A	F89AEF99	F79A05ED	A3DD4CC9
t=20	F33C0BD0	01DCFF0A	0904B856	7A88E68F	0CCD681F	0338AF7A	F89AEF99	F79A05ED
t=21	DE0E5BD2	F33C0BD0	01DCFF0A	0904B856	58974261	0CCD681F	0338AF7A	F89AEF99
t=22	43DEF3B8	DE0E5BD2	F33C0BD0	01DCFF0A	29512340	058974261	0CCD681F	0338AF7A
t=23	A6C10A92	43DEF3B8	DE0E5BD2	F33C0BD0	20E21E1E	29512340	058974261	0CCD681F

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
t=24	3B2C59E7	A6C10A92	43DEF3B8	DE0E5BD2	7B4EC01E	20E21E1E	29512340	58974261
t=25	649AA1E1	3B2C59E7	A6C10A92	43DEF3B8	2E34AE02	7B4EC01E	20E21E1E	29512340
t=26	40D33F39	649AA1E1	3B2C59E7	A6C10A92	21801FCF	2E34AE02	7B4EC01E	20E21E1E
t=27	3F047063	40D33F39	649AA1E1	3B2C59E7	A02914C0	21801FCF	2E34AE02	7B4EC01E
t=28	5AD16015	3F047063	40D33F39	649AA1E1	D4521FD4	A02914C0	21801FCF	2E34AE02
t=29	B50611F8	5AD16015	3F047063	40D33F39	AB3067C3	D4521FD4	A02914C0	21801FCF
t=30	A5713CD4	B50611F8	5AD16015	3F047063	EC7C3D02	AB3067C3	D4521FD4	A02914C0
t=31	1085B849	A5713CD4	B50611F8	5AD16015	8F2EC1AF	EC7C3D02	AB3067C3	D4521FD4
t=32	0339C020	1085B849	A5713CD4	B50611F8	187D18E0	8F2EC1AF	EC7C3D02	AB3067C3
t=33	91BDC1B2	0339C020	1085B849	A5713CD4	5EC331A0	187D18E0	8F2EC1AF	EC7C3D02
t=34	C9E98F7F	91BDC1B2	0339C020	1085B849	FE9FD46F	5EC331A0	187D18E0	8F2EC1AF
t=35	E605E2A0	C9E98F7F	91BDC1B2	0339C020	45180603	FE9FD46F	5EC331A0	187D18E0
t=36	835FCC9F	E605E2A0	C9E98F7F	91BDC1B2	89DEFDEE	45180603	FE9FD46F	5EC331A0
t=37	1DCEA9CE	835FCC9F	E605E2A0	C9E98F7F	F12504ED	89DEFDEE	45180603	FE9FD46F
t=38	AE5A5A99	1DCEA9CE	835FCC9F	E605E2A0	FD4F8519	F12504ED	89DEFDEE	45180603
t=39	8A6C6364	AE5A5A99	1DCEA9CE	835FCC9F	0AE1FA98	FD4F8519	F12504ED	89DEFDEE
t=40	AFFD88C5	8A6C6364	AE5A5A99	1DCEA9CE	1CDC1005	0AE1FA98	FD4F8519	F12504ED
t=41	8A6D5627	AFFD88C5	8A6C6364	AE5A5A99	CFBF9FFD	1CDC1005	0AE1FA98	FD4F8519
t=42	BD3A7AA7	8A6D5627	AFFD88C5	8A6C6364	FA27FA11	CFBF9FFD	1CDC1005	0AE1FA98
t=43	58E57D2D	BD3A7AA7	8A6D5627	AFFD88C5	603A9C5C	FA27FA11	CFBF9FFD	1CDC1005
t=44	5F1D2E5E	58E57D2D	BD3A7AA7	8A6D5627	4C060BF9	603A9C5C	FA27FA11	CFBF9FFD
t=45	5CE28408	5F1D2E5E	58E57D2D	BD3A7AA7	F88591FE	4C060BF9	603A9C5C	FA27FA11
t=46	5C73D167	5CE28408	5F1D2E5E	58E57D2D	FF5EB89D	F88591FE	4C060BF9	603A9C5C
t=47	1846719F	5C73D167	5CE28408	5F1D2E5E	8154DFC8	FF5EB89D	F88591FE	4C060BF9
t=48	752C7918	1846719F	5C73D167	5CE28408	24B7B432	8154DFC8	FF5EB89D	F88591FE
t=49	BC98B63A	752C7918	1846719F	5C73D167	58A7F232	24B7B432	8154DFC8	FF5EB89D
t=50	9E3712BA	BC98B63A	752C7918	1846719F	42734F4E	58A7F232	24B7B432	8154DFC8
t=51	5E4C9F13	9E3712BA	BC98B63A	752C7918	CC42FF0A	42734F4E	58A7F232	24B7B432
t=52	EBA56200	5E4C9F13	9E3712BA	BC98B63A	A53FAC05	CC42FF0A	42734F4E	58A7F232
t=53	7C4F010C	EBA56200	5E4C9F13	9E3712BA	9B5C9F2F	A53FAC05	CC42FF0A	42734F4E
t=54	E2757025	7C4F010C	EBA56200	5E4C9F13	D6EB6C15	9B5C9F2F	A53FAC05	CC42FF0A
t=55	EBD709CB	E2757025	7C4F010C	EBA56200	334B70AF	D6EB6C15	9B5C9F2F	A53FAC05
t=56	B06407F8	EBD709CB	E2757025	7C4F010C	C924B586	334B70AF	D6EB6C15	9B5C9F2F
t=57	C28EA6CC	B06407F8	EBD709CB	E2757025	D8A545D0	C924B586	334B70AF	D6EB6C15
t=58	A8743BE4	C28EA6CC	B06407F8	EBD709CB	3E627D09	D8A545D0	C924B586	334B70AF
t=59	78C36290	A8743BE4	C28EA6CC	B06407F8	9E5E6476	3E627D09	D8A545D0	C924B586
t=60	8104E8DD	78C36290	A8743BE4	C28EA6CC	41696EB5	9E5E6476	3E627D09	D8A545D0
t=61	969FE31D	8104E8DD	78C36290	A8743BE4	7BDBDB01	41696EB5	9E5E6476	3E627D09
t=62	6B3ED797	969FE31D	8104E8DD	78C36290	8063F67C	7BDBDB01	41696EB5	9E5E6476
t=63	F89E0D9C	6B3ED797	969FE31D	8104E8DD	5117DAD0	8063F67C	7BDBDB01	41696EB5

Untuk t=0 lakukan perhitungan sebagai berikut :

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
6A09E667	BB67AE85	3C6EF372	A54FF53A	510E527F	9B05688C	1F83D9AB	5BE0CD19

$$T_1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_t^{(256)} + W_t$$

$$h = 5\text{BE0CD19}$$

$$\sum_1^{(256)}(e) = (e\ ROTR\ 6) \oplus (e\ ROTR\ 11) \oplus (e\ ROTR\ 25)$$

$$\sum_1^{(256)}(e) = ((510E527F) ROTR\ 6) \oplus ((510E527F)\ ROTR\ 11) \oplus ((510E527F)\ ROTR\ 25)$$

$$\sum_1^{(256)}(e) = (FD443949) \oplus (4FEA21CA) \oplus (87293FA8)$$

$$\sum_1^{(256)}(e) = 3587272B$$

$Ch(e,f,g)$ = $(e\ \hat{}\ f) \oplus (\sim e\ \hat{}\ g)$

$Ch(e,f,g)$ = $(510E527F\ \hat{}\ 9B05688C) \oplus (\sim 510E527F\ \hat{}\ 1F83D9AB)$

$Ch(e,f,g)$ = $(1104400C) \oplus (0E818980)$

$Ch(e,f,g)$ = $1F85C98C$

$K_0^{(256)}$ = 428A2F98

W_t = 504B0304

T_1 = $5\text{BE0CD19} + 3587272B + 1F85C98C + 428\text{A2F98} + 504\text{B0304}$

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

T_1 =**43C2F06C**

$T_2 = \sum_0^{(256)} (a) + Maj(a,b,c)$

$\sum_0^{(256)} (a) = ((6A09E667)ROTR\ 2) \oplus ((6A09E667)ROTR\ 13) \oplus ((6A09E667)ROTR\ 22)$

$\sum_0^{(256)} (a) = (DA827999) \oplus (333B504F) \oplus (27999DA8)$

$\sum_0^{(256)} (a) = CE20B47E$

$Maj(a,b,c)$ = $(a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$

$Maj(a,b,c)$ = $(6A09E667 \wedge BB67AE85) \oplus (6A09E667 \wedge 3C6EF372) \oplus (BB67AE85 \wedge 3C6EF372)$

$Maj(a,b,c)$ = $(2A01A605) \oplus (2808E262) \oplus (0866A200)$

$Maj(a,b,c)$ = $0A6FE667$

$T_2 = CE20B47E + 0A6FE667$

$T_2 = \boldsymbol{D8909AE5}$

h =g
=1F83D9AB

g =f
= 9B05688C

f =e
= 510E527F

e =$d + T_1$
=A54FF53A + 43C2F06C
= E912E5A6

d =c
=3C6EF372

c =b
=BB67AE85

b =a
=6A09E667

a =$T_1 + T_2$
=43C2F06C + D8909AE5
= 1C538B51

Untuk t=1 lakukan perhitungan sebagai berikut :

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
1C538B51	6A09E667	BB67AE85	3C6EF372	E912E5A6	510E527F	9B05688C	1F83D9AB

$T_1 = h + \sum_1^{(256)} (e) + Ch(e,f,g) + K_t^{(256)} + W_t$

h = 1F83D9AB

$\sum_1^{(256)} (e) = (e\ ROTR\ 6) \oplus (e\ ROTR\ 11) \oplus (e\ ROTR\ 25)$

$\sum_1^{(256)} (e) = ((E912E5A6)ROTR\ 6) \oplus ((E912E5A6)\ ROTR\ 11) \oplus ((E912E5A6)\ ROTR\ 25)$

$\sum_1^{(256)} (e) = (9BA44B96) \oplus (B4DD225C) \oplus (8972D374)$

$\sum_1^{(256)} (e) = A60BBABE$

$Ch(e,f,g)$ = $(e \wedge f) \oplus (\sim e \wedge g)$

$Ch(e,f,g)$ = $(E912E5A6 \wedge 510E527F) \oplus (\sim E912E5A6 \wedge 9B05688C)$

$Ch(e,f,g)$ = $(41024026) \oplus (72178D2A)$

$Ch(e,f,g)$ = $3315CD0C$

$K_0^{(256)}$ = 71374491

W_t = 14000600

T_1 = $1F83D9AB + A60BBABE + 3315CD0C + 71374491 + 14000600$

T_1 =**7DDCAC06**

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

$T_2 = \sum_0^{(256)} (a) + Maj(a,b,c)$
$\sum_0^{(256)} (a) = ((1C538B51)ROTR\ 2) \oplus ((1C538B51)ROTR\ 13) \oplus ((1C538B51)ROTR\ 22)$
$\sum_0^{(256)} (a) = (4714E2D4) \oplus (5A88E29C) \oplus (4E2D4471)$
$\sum_0^{(256)} (a) = 53B14439$
$Maj(a,b,c)$ = $(a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
$Maj(a,b,c)$ = $(1C538B51 \wedge 6A09E667) \oplus (1C538B51 \wedge BB67AE85) \oplus (6A09E667 \wedge BB67AE85)$
$Maj(a,b,c)$ = $(08018241) \oplus (18438A01) \oplus (2A01A605)$
$Maj(a,b,c)$ = $3A43AE45$
$T_2 = 53B14439 + 3A43AE45$
$T_2 = \mathbf{8DF4F27E}$
h =g
=9B05688C
g =f
= 510E527F
f =e
= E912E5A6
e =$d + T_1$
=3C6EF372 + 7DDCAC06
= BA4B9F78
d =c
=BB67AE85
c =b
=6A09E667
b =a
=1C538B51
a =$T_1 + T_2$
=7DDCAC06 + $8DF4F27E$
= 1BE9E4FC

Untuk t=2 lakukan perhitungan sebagai berikut :

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
1BE9E4FC	1C538B51	6A09E667	BB67AE85	BA4B9F78	E912E5A6	510E527F	9B05688C

$T_1 = h + \sum_1^{(256)} (e) + Ch(e,f,g) + K_t^{(256)} + W_t$
h = 9B05688C

$\sum_1^{(256)} (e) = (e\ ROTR\ 6) \oplus (e\ ROTR\ 11) \oplus (e\ ROTR\ 25)$
$\sum_1^{(256)} (e) = ((BA4B9F78)ROTR\ 6) \oplus ((BA4B9F78)\ ROTR\ 11) \oplus ((BA4B9F78)\ ROTR\ 25)$
$\sum_1^{(256)} (e) = (E2E92E7D) \oplus (EF174973) \oplus (25CFBC5D)$
$\sum_1^{(256)} (e) = 2831DB53$

$Ch(e,f,g)$ = $(e \wedge f) \oplus (\sim e \wedge g)$
$Ch(e,f,g)$ = $(BA4B9F78 \wedge E912E5A6) \oplus (\sim BA4B9F78 \wedge 510E527F)$
$Ch(e,f,g)$ = $(A8028520) \oplus (41044007)$
$Ch(e,f,g)$ = $E906C527$
$K_0^{(256)}$ = B5C0FBCF

W_t = 08000000
T_1 = 9B05688C + $2831DB53$ + $E906C527$ + B5C0FBCF + 08000000
T_1 =**69FF04D5**

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

$T_2 = \sum_0^{(256)}(a) + Maj(a,b,c)$
$\sum_0^{(256)}(a) = ((1BE9E4FC)ROTR\ 2) \oplus ((1BE9E4FC)ROTR\ 13) \oplus ((1BE9E4FC)ROTR\ 22)$
$\sum_0^{(256)}(a) = (06FA793F) \oplus (27E0DF4F) \oplus (A793F06F)$
$\sum_0^{(256)}(a) = 8689561F$

$Maj(a,b,c)$ = $(a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
$Maj(a,b,c)$ = $(1BE9E4FC \wedge 1C538B51) \oplus (1BE9E4FC \wedge 6A09E667) \oplus (1C538B51 \wedge 6A09E667)$
$Maj(a,b,c)$ = $(18418050) \oplus (0A09E464) \oplus (08018241)$
$Maj(a,b,c)$ = $1A449E675$

$T_2 = 8689561F9 + 1A449E675$
T_2 = **A0D33C94**
h =g
=510E527F
g =f
= E912E5A6
f =e
= BA4B9F78
e =$d + T_1$
=BB67AE85 + 69FF04D5
= 2566B35A
d =c
=6A09E667
c =b
=1C538B51
b =a
=1BE9E4FC
a =$T_1 + T_2$
=69FF04D5 + A0D33C94
= 0AD24169

Untuk t=3 lakukan perhitungan sebagai berikut :

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
0AD24169	1BE9E4FC	1C538B51	6A09E667	2566B35A	BA4B9F78	E912E5A6	510E527F

$T_1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_t^{(256)} + W_t$
h = 510E527F

$\sum_1^{(256)}(e) = (e\ ROTR\ 6) \oplus (e\ ROTR\ 11) \oplus (e\ ROTR\ 25)$
$\sum_1^{(256)}(e) = ((2566B35A)ROTR\ 6) \oplus ((2566B35A)\ ROTR\ 11) \oplus ((2566B35A)\ ROTR\ 25)$
$\sum_1^{(256)}(e) = (68959ACD) \oplus (6B44ACD6) \oplus (B359AD12)$
$\sum_1^{(256)}(e) = B0889B09$

$Ch(e,f,g)$ = $(e \wedge f) \oplus (\sim e \wedge g)$
$Ch(e,f,g)$ = $(2566B35A \wedge BA4B9F78) \oplus (\sim 2566B35A \wedge E912E5A6)$
$Ch(e,f,g)$ = $(20429358) \oplus (C81044A4)$
$Ch(e,f,g)$ = $E852D7FC$
$K_0^{(256)}$ = E9B5DBA5

W_t = 2100007B

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

T_1 = 510E527F + $B0889B09$ + $E852D7FC$ + E9B5DBA5 + 2100007B
T_1 =**F49FA1A4**

$T_2 = \sum_0^{(256)} (a) + Maj(a,b,c)$
$\sum_0^{(256)} (a) = ((0\text{AD}24169)ROTR\ 2) \oplus ((0\text{AD}24169)ROTR\ 13) \oplus ((0\text{AD}24169)ROTR\ 22)$
$\sum_0^{(256)} (a) = (42B4905A) \oplus (0B485692) \oplus (4905A42B)$
$\sum_0^{(256)} (a) = 00F962E3$

$Maj(a,b,c)$ = $(a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
$Maj(a,b,c)$ = $(0\text{AD}24169 \wedge 1\text{BE9E4FC}) \oplus (0\text{AD}24169 \wedge 1\text{C538B51}) \oplus (1\text{BE9E4FC} \wedge 1\text{C538B51})$
$Maj(a,b,c)$ = $(0AC04068) \oplus (08520141) \oplus (18418050)$
$Maj(a,b,c)$ = $1AD3C179$

$T_2 = 00F962E3 + 1AD3C179$
T_2 = **1BCD245C**
h =g
=E912E5A6
g =f
= BA4B9F78
f =e
= 2566B35A
e =$d + T_1$
=6A09E667 + F49FA1A4
= 5EA9880B
d =c
=1C538B51
c =b
=1BE9E4FC
b =a
=0AD24169
a =$T_1 + T_2$
= F49FA1A4 + 1BCD245C
= 106CC600

Setelah mendapatkan hasil perhitungan t=63 maka di dapatkan hasil dari nilai hash sebagai berikut :

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
F89E0D9C	6B3ED797	969FE31D	8104E8DD	5117DAD0	8063F67C	7BDBDB01	41696EB5

7. Menghitung *IntermediateHashValue*

$H_0^{(1)} = a + H_0^{(1-1)}$
$H_0^{(1)}$ = F89E0D9C + 6A09E667
$H_0^{(1)} =$ **62A7F403**

$H_1^{(1)} = b + H_0^{(1-1)}$
$H_1^{(1)}$ = 6B3ED797 + BB67AE85
$H_1^{(1)} =$ **26A6861C**

$H_2^{(1)} = c + H_0^{(1-1)}$
$H_2^{(1)}$ = 969FE31D + $3C6EF372$
$H_2^{(1)} =$ **D30ED68F**

$H_3^{(1)} = d + H_0^{(1-1)}$
$H_3^{(1)}$ = 8104E8DD + A54FF53A
$H_3^{(1)} =$ **2654DE17**

ISSN 2745-7966 (Media Online)
Vol 2, No 5, Mei 2022
Hal 201-213
https://djournals.com/resolusi

$H_4^{(1)} = e + H_0^{(1-1)}$
$H_4^{(1)} = 5117DAD0 + 510E527F$
$H_4^{(1)} = \mathbf{A2262D4F}$

$H_5^{(1)} = f + H_0^{(1-1)}$
$H_5^{(1)} = 8063F67C + 9B05688C$
$H_5^{(1)} = \mathbf{1B695F08}$

$H_6^{(1)} = g + H_0^{(1-1)}$
$H_6^{(1)} = 7BDBDB01 + 1F83D9AB$
$H_6^{(1)} = \mathbf{9B5FB4AC}$

$H_7^{(1)} = h + H_0^{(1-1)}$
$H_7^{(1)} = 41696EB5 + 5BE0CD19$
$H_7^{(1)} = \mathbf{9D4A3BCE}$

8. Output
   Output dari SHA-256 merupakan penggabungan dari *H0(0)* sampai *H7(0)* sebagai berikut :
   62A7F403 || 26A6861C || D30ED68F || 2654DE17 || A2262D4F || 1B695F08 || 9B5FB4AC || 9D4A3BCE

   Sehingga didapat nilai *hash*dari pesan M adalah sebagai berikut :

   62A7F40326A6861CD30ED68F2654DE17A2262D4F1B695F089B5FB4AC9D4A3BCE

4. KESIMPULAN

Berdasarkan pembahasan dan penulisan dari bab-bab sebelumnya, maka dapat diambil kesimpulan, dimana kesimpulan tersebut kiranya dapat berguna. Adapun kesimpulan-kesimpulan tersebut Dalam mencari file dokumen dapat memberikan identitas dari setiap *file* dan melakukan *scanning* pada sebuah ruang penyimpanan yang di dalamnya terdapat banyak *file* dokumen. Dengan menerapkan algoritma SHA-256 dapat membangkitkan identitas dari *file* dokumen hasil *scanning* tersebut. Melakukan pengelompokkan *file* dokumen berdasarkan identitas *file* dokumen yang di bangkitkan menggunakan fungsi *hash* SHA-256.

REFERENCES

[1] A. G. Tammam, "Fungsi Hash dan algoritma SHA-256," Keamanan Komputer, 2016.
[2] S. M. Egi Cahyo Prabowo Irawan Afrianto, "Implementasi Aplikasi Digital Signature Menggunakan Fungsi Hash Algoritma SHA-256 dan RSA di Badan Pertanahan Nasional Kota Cimahi," Teknik Informatika, pp. 1-8, 2014.
[3] Imam Saputra, "Analisa Algoritma SHA-256 Untuk Mendeteksi Orisinalitas Citra Digital," in Seminar Nasional Riset Information Science (SENARIS), Pematang Siantar, 2019.
[4] S.J Patil, N.P Jagtap, S.H Rajput, and R.B Sangore, "A Duplicate File Finder System," International Journal of Science Spirituality Business and Technology, pp. 10-14, 2017.
[5] Whitfield Diffie and Martin E Hellman, "New Direction in Cryptography," vol. IT 22, no. 6, November 1976.
[6] Dony Ariyus , PengantarIlmu Kriptografi. Yogyakarta: Andi, 2008.
[7] R Sadikin, Kriptografi keamanan jaringan dan implementasinya dalam bahasa java. Yogyakarta: CV Andi Offset, 2012.
[8] Prayudi , Yudi , and Idham Halik, "Studi Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data," Seminar Nasional Aplikasi Teknologi Informasi 2005(SNATI 2005), 2005.
[9] M Syafriadi, "Analisis Kecepatan dan Keamanan Algoritma Secure Hash Algorithm 256 (SHA-256) Untuk Otentikasi Pesan Teks," 2006.
[10] Booch , G James, and R Ivar, The Unified Modeling Language User Guide Second Edition. United State: Addison Wesley profesional, 2005.
[11] Adi Nugroho, Rekayasa Perangkat Lunak Menggunakan UML dan Java. Yogyakarta: Andi Offset, 2010.
[12] A.S Rosa and M Shalahuddin, Rekayasa Perangkat Lunak Struktur dan berorientasi Objek. Bandung: Informatika, 2014.
[13] Rahmat Priyanto, Lansung bisa Visual Basic Net 2008. Yogyakarta: Andi, 2009.
[14] Nurdin Usman,. Bandung: CV sinar baru, 2002.