JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169

Perancangan Aplikasi Duplicate Document Scanner Menerapkan Algoritma SHA 1

Nur Indah Utami*, Berto Nadeak, Imam Saputra

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: nurindahu13@gmail.com
Submitted **12-09-2021**; Accepted **20-10-2021**; Published **30-10-2021**

Abstrak

Saat ini kita sudah berada dalam era dunia digital. Banyak industri yang sudah beralih ke teknologi digital seperti foto, video, musik, dan sebagainya. Tentunya perubahan ini membawa banyak manfaat untuk kita dalam hal kecepatan dan akses data. Hal ini berdampak pada banyaknya file yang sama ataupun file duplikat yang tersimpan dalam satu komputer, hal ini tentunya tidak diperlukan dan menyita ruang penyimpanan. Sama halnya dengan file dokumen, akan sangat sulit untuk membedakan file dokumen tanpa melihat isi dari file dokumen tersebut. Dalam kriptografi terdapat fungsi hash yang merupakan fungsi satu arah yang dapat membangkitkan identitas sebuah file, dimana jika file dokumen itu isinya sama maka akan menghasilkan nilai hash yang sama pula. Hal ini dapat digunakan untuk mengidentifikasi file dokumen yang sama ataupun duplikat.Dalam fungsi hash terdapat metode SHA-1 ataupun Secure Hash Algorithm 1 yang merupakan salah satu algoritma hashing yang sering digunakan untuk mengenkripsi data dengan lebar 20 byte atau 160 bit.

Kata Kunci: Scan; Perancangan; Algoritma SHA-1; Document

Abstract

We are now in the era of the digital world. Many industries have switched to digital technology such as photos, videos, music, and so on. Of course, this change brings many benefits for us in terms of speed and data access. This has an impact on the number of the same files or duplicate files stored on one computer, this is certainly not needed and takes up storage space. As with document files, it will be very difficult to distinguish document files without seeing the contents of the document file. In cryptography there is a hash function which is a one-way function that can generate the identity of a file, where if the document file contains the same content, it will produce the same hash value. This can be used to identify the same or duplicate document files. In the hash function there is the SHA-1 method or the Secure Hash Algorithm 1 which is one of the hashing algorithms that is often used to encrypt data with a width of 20 bytes or 160 bits.

Keywords: Scan; Design; SHA-1 Algorithm; Documents

1. PENDAHULUAN

File dengan tipe document menyimpan banyak informasi yang di sajikan dalam bentuk karakter maupun gambar yang dapat diakses dengan mudah melalui aplikasi word processing. File ini juga banyak tersedia di berbagai server dan dapat di unduh dengan mudah oleh pengguna. Karena kemudahan pengunduhan ini maka pengguna tidak sadar bahwa telah mengunduh file document yang sama tetapi dengan nama yang berbeda ataupun disimpan dalam ruang penyimpanan yang berbeda. Hal ini tentunya akan membuat ruang penyimpanan penuh dengan file document yang memiliki isi yang sama dan menyita ruang penyimpanan.

Untuk dapat membedakan antara file document yang satu dengan yang lainnya maka pengguna perlu untuk membuka dan membaca isi dari file document secara keseluruhan. Hal ini tentunya sangat merepotkan dan membutuhkan waktu yang lama dan ketelitian yang tinggi apalagi jika isi file document tersebut tidak memiliki perbedaan yang signifikan. Maka dibutuhkan sebuah solusi yang dapat digunakan untuk membedakan satu file document yang satu dengan file document yang lainnya yang dapat dilakukan dengan mudah dan cepat. Sehingga dapat digunakan oleh pengguna dalam mengambil keputusan untuk menghapus file document yang ganda atau duplikat. Sehingga dapat menghemat ruang penyimpanan yang digunakan.

Dalam kriptografi terdapat fungsi hash yang merupakan fungsi satu arah yang dapat membangkitkan identitas sebuah file, dimana jika file dokumen itu isinya sama maka akan menghasilkan nilai hash yang sama pula. Hal ini dapat digunakan untuk mengidentifikasi file dokumen yang sama atau pun duplikat. Algoritma SHA-1 memiliki input panjang string yang berbeda akan menghasilkan output dengan panjang string yang tetap 160 bit. Penerapan algoritma SHA-1 dapat dibuat dalam bentuk library pada bahasa java sehingga dapat langsung diimplementasikan untuk otentikasi atau pengamanan data maupun untuk pembuatan tanda tangan digital [1]. Algoritma SHA-1 unggul dalam pengujian avalanche effect yaitu perubahan 1 bit pada string input menyebabkan perubahan 100 persen pada string output dan SHA-1 memiliki waktu pemrosesan yang lebih cepat dibandingkan dengan varian Secure Hash Algorithm lainnya [2]. Dengan menerapkan algoritma ini pada aplikasi file scanner maka hasil dari pencarian file dokumen tersebut di kelompokkan berdasarkan nilai hash yang sama dan pengguna dapat memudahkan dan mempercepat pengguna untuk menghapus file dokumen yang duplikat tanpa harus membuka dan membaca isi file tersebut

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Steganografi adalah ilmu menyembunyikan pesan tersembunyi dengan suatu cara sehingga selain si pengirim dan penerima, tidak ada seorang pun yang mengetahui atau menyadari keberadaan suatu pesan rahasia. Steganografi berbeda

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169

dengan watermarking dan kriptografi yang menyamarkan arti pesan, namun tidak menyembunyikan keberadaan pesan. Steganografi dan kriptografi sering digunakan secara bersamaan untuk menjamin keamanan pesan rahasia.Encryption merupakan proses penyamaran sebuah pesan yang bertujuan agar isi pesan tersebut hanya diketahui oleh pihak yang berhak. Decryption merupakan pengembalian pesan menjadi kondisi aslinya. Integrity memverifikasi bahwa pesan yang dikirim oleh pengirim kepada penerima tidak mengalami perubahan saat pengiriman sedang berlangsung. Non-repudiation menghindari penyangkalan bahwa pihak tertentu telah mengirimkan pesan atau menerima pesan.Kriptografi klasik umumnya merupakan teknik penyandian dengan kunci simetris dan pesan disembunyikan sehingga tidak memiliki arti dengan metode substitusi dan/atau transposisi[1][3][4].

2.2 Fungsi Hash

Fungsi hash adalah sebuah fungsi yang menerima masukan string yang panjangnya sembarang dan mengkonversikannya menjadi string keluaran yang panjangnya tetap dan umumnya jauh lebih kecil dibandingkan string semula. Fungsi hash satu arah adalah fungsi hash yang bekerja dalam satu arah. Pesan yang sudah diubah menjadi message digest tidak dapat lagi dikembalikan menjadi pesan semula. Dua pesan yang berbeda akan selalu menghasilkan nilai hash yang berbeda pula.Secure Hash Algorithm dikembangkan oleh NSA pada tahun 1995 dan distandarisasi oleh NIST . SHA-1 merupakan algoritma hash searah dengan panjang maksimal untuk suatu string yang dapat di proses adalah 264 bit. SHA-1 akan menghasilkan keluaran sebanyak 160 bit dari string tersebut [5][6].

2.3 *Duplicate Document Scanner*

Duplicate document scanner adalah proses pencarian file yang duplikat atau ganda sebagian besar menggunakan algoritma fungsi hash standar . Algoritma fungsi hash bertindak sebagai enkripsi satu arah dan menghasilkan identitas dari sebuah file dokumen dan setiap file dokumen menghasilkan nilai hash yang unik.Pencarian jenis ini akan mencari file dengan nama, ekstensi dan ukuran dengan panjang checksum 128 bit.Pencarian ini dilakukan dengan mencocokkan byte demi byte dan dengan mengambil sampel yang berbeda dalam sebuah file dan membandingkannya dengan sampel file yang lain [7].

2.4 File Dokumen

File DOC merupakan dokumen pengolah kata yang diciptakan oleh program Microsoft Word. Jenis file ini bisa berisi susunan teks, image, tabel, grafik, chart, format halaman dan pengaturan cetak. Setiap file DOC selalu diikuti dengan ekstensi .doc. Jenis file DOC secara otomatis akan tercipta ketika membuat dokumen pada Microsoft Word dan menyimpannya.Namun, untuk mengetahui identitas atau jenis file tersebut, bisa dilihat dari kolom tipe di mana pada kolom tersebut diinformasikan jenis file tersebut adalah Microsoft Word Document. Sebagai file dokumen, file DOC bisa dibuka secara langsung dan diedit sedemikian rupa menggunakan program-program tertentu.

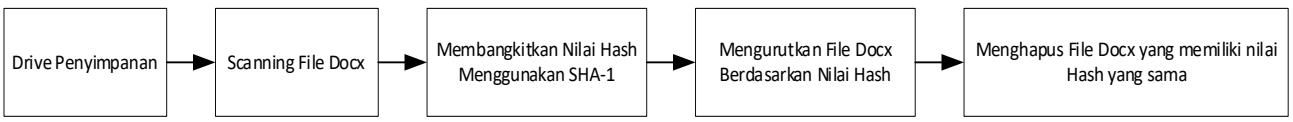
2.5 Mengunduh atau Menyimpan file

Pada saat kita mengunduh suatu file terkadang kita sering mengalami masalah seperti masalah jaringan yang membuat proses pengunduhan yang kita lakukan menjadi gagal atau masalah ukuran atau kapasitas yang kita unduh terlalu besar. Tanpa kita sadari, proses yang kita lakuan itu berdampak pada kapasitas penyimpanan komputer kita yang berkurang.File document hasil unduhan kita yang sama akan menyita ruang penyimpanan kita bila tidak di hapus. Maka dari itu dibutuhkan sebuah aplikasi yang dapat digunakan untuk mencari document tersebut agar pengguna dapat mengambil keputusan untuk menghapus file document yang ganda atau duplikat[8].

3. HASIL DAN PEMBAHASAN

3.1 Pembahasan

Karena di zaman sekarang sangat mudah untuk mengambil atau mengunduh suatu file maka terkadang pengguna tidak sadar telah mengunduh file document yang sama tetapi dengan nama yang berbeda ataupun disimpan dalam ruang penyimpanan yang berbeda. Maka pengguna membutuhkan suatu aplikasi duplicate document scanner yang dapat digunakan untuk menyelesaikan semua masalah itu secara cepat dan mudah.Algoritma fungsi hash bertindak sebagai enkripsi satu arah dan menghasilkan identitas dari sebuah file dokumen dan setiap file dokumen menghasilkan nilai hash yang unik. Untuk mendeteksi file dokumen yang duplikat atau ganda maka di lakukan perbandingan antara nilai hash dari masing-masing file dokumen. Untuk lebih jelasnya dapat di lihat pada gambar 1. di bawah ini:



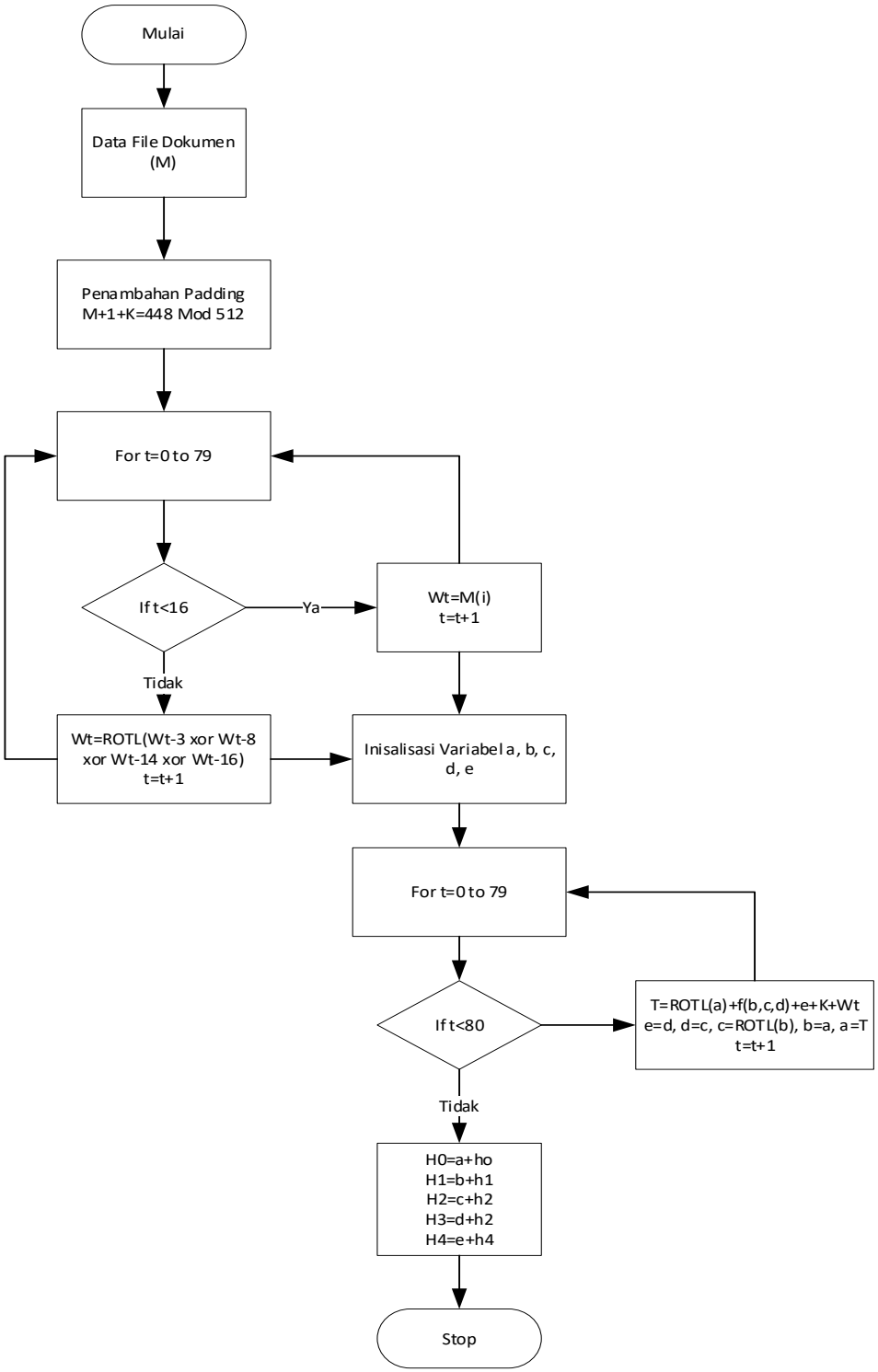
Gambar 1. Alur Proses Scanning File DOCX

Algoritma SHA-1 merupakan algoritma fungsi *hash* yang menggunakan *input* sepanjang 512 bit dan menghasilkan *output* 160 bit. Penerapan algoritma SHA-1 pada aplikasi Duplicate Document Scanner adalah untuk membangkitkan identitas setiap *file document* yang ada pada sebuah media penyimpanan. Langkah yang dilakukan adalah dengan melakukan

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169

scanning ataupun pencarian *file document* yang berekstensi docx, selanjutnya membangkitkan nilai *hash* dari setiap *file* docx tersebut, setelah setiap *file* docx tersebut memiliki nilai *hash* masing-masing maka *file* docx tersebut akan di urutkan berdasarkan nilai *hash*nya dan pengguna dapat menghapus *file* docx yang memiliki nilai hash yang sama karena *file* docx tersebut merupakan *file* docx yang sama atau duplikat

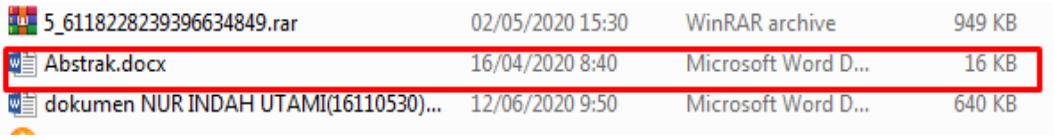
Dalam penelitian ini algoritma yang digunakan untuk memecahkan masalah pencarian *file* dokumen yang duplikat atau ganda adalah algoritma SHA-1. Visualisasi algoritma dapat berupa *flowchart* ataupun dalam bentuk *pseudocode.* Dalam penelitian ini digunakan flowchart dalam memvisualisasikan algoritma SHA-1. Algoritma SHA-1 ini dapat di lihat pada gambar 4. berikut ini:



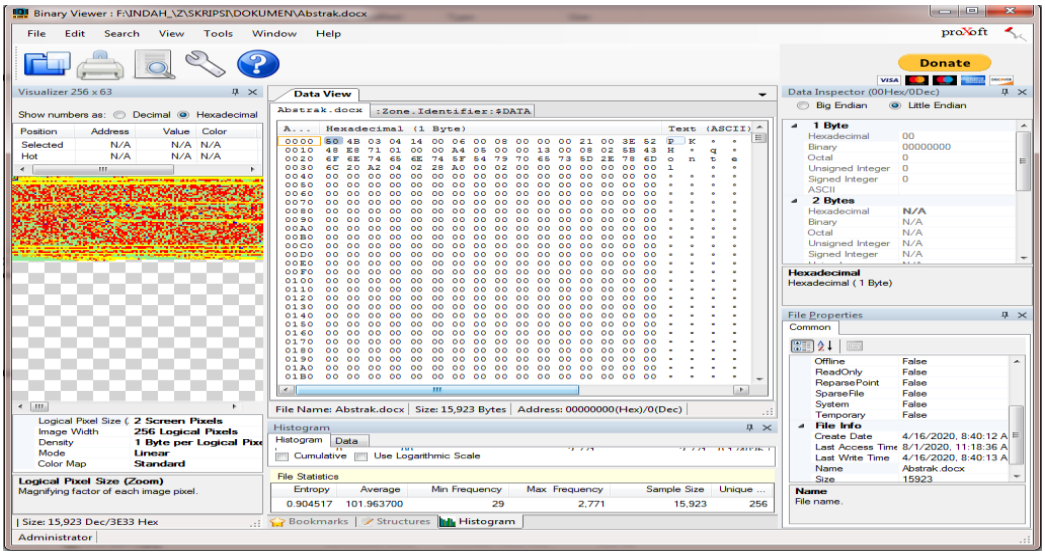
Gambar 2. Flowchart Algoritma SHA-1

Dalam contoh kasus ini pengguna akan menggunakan file document berekstensi docx dengan ukuran file 16 KB dan menggunakan aplikasi bantuan *binary viewer* untuk mendapatkan bilangan hexadecimal dari file tersebut.

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169



Gambar 2. File Document



Gambar 3. Aplikasi *Binary Viewers*

Tabel 1 Heksadesimal 5 x 5

50	4B	03	04	14
00	06	00	08	00
00	00	21	00	3E
52	48	E8	71	01
00	00	A4	05	00

Pesan : 504B030414, 0006000800, 000021003E, 5248E87101, 0000A40500

Tabel 2. Input nilai heksadesimal ke biner

01010000	01001011	00000011	00000100	00010100
00000000	00000110	00000000	00001000	00000000
00000000	00000000	00100001	00000000	00111110
01010010	01001000	11101000	01110001	00000001
00000000	00000000	10100100	00000101	00000000

Pesan yang di atas diubah menjadi biner dan proses perhitungan menggunakan metode SHA-1 dari 0 – 79 putaran. Langkah pertama yaitu :

1. Melakukan Penambahan *padding bit.*
   200 + 1 + K = 448 mod 512
   K = 448 – 201 mod 512
   K = 247
   Lakukan penambahan bit 1 kemudian diikuti oleh bit 0 sebanyak 247 bit.

Tabel 3. Penambahan Padding 247 Bit

M0	01010000	01001011	00000011	00000100
M1	00010100	00000000	00000110	00000000
M2	00001000	00000000	00000000	00000000
M3	00100001	00000000	00111110	01010010
M4	01001000	11101000	01110001	00000001
M5	00000000	00000000	10100100	00000101
M6	00000000	**10000000**	**00000000**	**00000000**
M7	**00000000**	**00000000**	**00000000**	**00000000**

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169

M8	**00000000**	**00000000**	**00000000**	**00000000**
M9	**00000000**	**00000000**	**00000000**	**00000000**
M10	**00000000**	**00000000**	**00000000**	**00000000**
M11	**00000000**	**00000000**	**00000000**	**00000000**
M12	**00000000**	**00000000**	**00000000**	**00000000**
M13	**00000000**	**00000000**	**00000000**	**00000000**

Biner yang dihitamkan dalam tabel di atas adalah nilai dari tambahan bit sebanyak 247 bit.

2. Penambahan panjang pesan
Melakukan penambahan panjang pesan sebanyak 64 bit dan dibuat juga seperti tabel di atas dan tambahan nilainya dihitamkan supaya diketahui mana yang bit ditambahkan.

Tabel 4. Penambahan panjang pesan

M0	01010000	01001011	00000011	00000100
M1	00010100	00000000	00000110	00000000
M2	00001000	00000000	00000000	00000000
M3	00100001	00000000	00111110	01010010
M4	01001000	11101000	01110001	00000001
M5	00000000	00000000	10100100	00000101
M6	00000000	10000000	00000000	00000000
M7	00000000	00000000	00000000	00000000
M8	00000000	00000000	00000000	00000000
M9	00000000	00000000	00000000	00000000
M10	00000000	00000000	00000000	00000000
M11	00000000	00000000	00000000	00000000
M12	00000000	00000000	00000000	00000000
M13	00000000	00000000	00000000	00000000
M14	**00000000**	**00000000**	**00000000**	**00000000**
M15	**00000000**	**00000000**	**00000000**	**10001000**

3. *Parsing* pesan (Mengelompokan Pesan)
Melakukan pengelompokan pesan dari penambahan bit sampai ke penambahan panjang pesan.

Tabel 5. Pengelompokkan pesan

M0	01010000	01001011	00000011	00000100	=504B0304
M1	00010100	00000000	00000110	00000000	=14000600
M2	00001000	00000000	00000000	00000000	=08000000
M3	00100001	00000000	00111110	01010010	=21003E52
M4	01001000	11101000	01110001	00000001	=48E87101
M5	00000000	00000000	10100100	00000101	=0000A405
M6	00000000	10000000	00000000	00000000	=00800000
M7	00000000	00000000	00000000	00000000	=00000000
M8	00000000	00000000	00000000	00000000	=00000000
M9	00000000	00000000	00000000	00000000	=00000000
M10	00000000	00000000	00000000	00000000	=00000000
M11	00000000	00000000	00000000	00000000	=00000000
M12	00000000	00000000	00000000	00000000	=00000000
M13	00000000	00000000	00000000	00000000	=00000000
M14	00000000	00000000	00000000	00000000	=00000000
M15	00000000	00000000	00000000	10001000	=00000000

4. Penjadwalan Pesan
Melakukan penjadwalan pesan, langkah ini diawali dengan mengubah setiap blok data menjadi bilangan heksadesimal dengan menggunakan rumus :

$$Wt = \begin{cases} M_t^{(t)} & 0 \le t \le 15 \\ ROTL^{(1)}(W_{t-3}) \oplus (W_{t-8}) \oplus (W_{t-14}) \oplus (W_{t-16}) & 16 \le t \le 79 \end{cases} \quad (1)$$

Tabel 6. Penjadwalan pesan

504B0304	14000600	08000000	21003E52	48E87101	0000A405	00800000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
B0960608	6A0070A4	81D0E202	232D38BF	44D0034A	03A08C0F	475A717E	89A0034A
666D140F	5AB403B4	10E1C290	8A805960	3CC801FC	26829D3E	FA985C2C	BED0E425

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169

E352DA76	D2022F07	D6638FFA	92BE6F4C	DA80B99F	42876398	9222AC22	6E88BA89
AEB2DEE3	2029B3E3	2985EC3F	201CEBE3	79F2AF58	B8E40190	57E98337	F751BDE6
46CF15FE	6EF8E13E	A46ECF79	6DD5E593	BF50B7B6	610A2FF0	0D5828CA	0D400674
41E411A5	C72B23E8	F2B314D5	69939C8B	ACC17062	B818AEA7	EBDA4510	7050559C
36BACB0A	FAB260E1	33BA730D	A7ED3BC4	C8F71FFF	CFD1E85D	01168F76	9110C9AA
94784C4E	AA38A1E8	F9B1BC21	C43C8B4D	4BA916CA	FC515E8F	30951442	A0B7D43A

Untuk menjadwalkan data ke 16 sampai 79 dilakukan perhitungan sebagai berikut:
Data ke 16

$Wt = ROTL^{(1)}(W_{t-3}) \oplus (W_{t-8}) \oplus (W_{t-14}) \oplus (W_{t-16})$
$W_{16} = ROTL^{(1)}(W_{16-3}) \oplus (W_{16-8}) \oplus (W_{16-14}) \oplus (W_{16-16})$
$W_{16} = ROTL^{(1)}(W_{13}) \oplus (W_8) \oplus (W_2) \oplus (W_0)$
$W_{16} = ROTL^{(1)}(00000000) \oplus (00000000) \oplus (08000000) \oplus (504B0304)$
$W_{16} = ROTL^{(1)}(584B0304)$
$W_{16} = B0960608$

Perhitungan di kerjakan hingga data ke 79

5. Inisialisasi Variabel Kerja
   Setelah melakukan penjadwalan pesan, maka langkah selanjutnya adalah inisialisasi nilai *hash* di mana nilai ini merupakan sebuah ketentuan yaitu:

Tabel 7. Inisial *hash value*

$H_0^{(0)}$	$H_1^{(0)}$	$H_2^{(0)}$	$H_3^{(0)}$	$H_4^{(0)}$
67452301	EFCDAB89	98BADCFE	10325476	C3D2E1F0

6. Proses Variabel Kerja
   T=0

a	*b*	*c*	*d*	*e*
67452301	EFCDAB89	98BADCFE	10325476	C3D2E1F0

$T = ROTL^{(5)}(a) + f_0(b,c,d) + e + K_0 + W_0$
$f_0(b,c,d)$ = $(b \wedge c) \oplus (\sim b \wedge d)$
$f_0(b,c,d)$ = $(EFCDAB89 \wedge 98BADCFE) \oplus (\sim EFCDAB89 \wedge 10325476)$
$f_0(b,c,d)$ = $(88888888) \oplus (10325476)$
$f_0(b,c,d)$ = $98BADCFE$

$T = ROTL^{(5)}(67452301) + (98BADCFE) + (C3D2E1F0) + (5A827999) + (504B0304)$
$T = (E8A4602C) + (98BADCFE) + (C3D2E1F0) + (5A827999) + (504B0304)$
$\boldsymbol{T = (EFFF9BB7)}$
$\boldsymbol{e = (10325476)}$
$\boldsymbol{d = (98BADCFE)}$
$c = ROTL^{(30)}(EFCDAB89)$
$\boldsymbol{c = (7BF36AE2)}$
$\boldsymbol{b = (67452301)}$
$\boldsymbol{a = (EFFF9BB7)}$

Proses variable kerja di kerjakan hingga T=79

7. Menghitung nilai *hash* ke i
$H_0^{(1)} = a + H_0^{(1-1)}$
$H_0^{(1)} = BB0247A6 + 67452301 + H_0^{(1-1)}$
$H_0^{(1)} = 22476AA7 + H_0^{(1-1)}$
$H_1^{(1)} = b + H_1^{(1-1)}$
$H_1^{(1)} = 505F6A9D + EFCDAB89 + H_0^{(1-1)}$
$H_1^{(1)} = 402D1626$
$H_2^{(1)} = c + H_2^{(1-1)}$
$H_2^{(1)} = C10C1097 + 98BADCFE + H_0^{(1-1)}$
$H_2^{(1)} = 59C6ED96$
$H_3^{(1)} = d + H_3^{(1-1)}$
$H_3^{(1)} = 1D8BCE15 + 10325476 + H_0^{(1-1)}$
$H_3^{(1)} = 2DBE228B$
$H_4^{(1)} = e + H_4^{(1-1)}$
$H_4^{(1)} = 273F6250 + C3D2E1F0 + H_0^{(1-1)}$

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 5, Oktober 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v8i5.3627
Hal 163−169

$H_4^{(1)} = EB124440$

8. *Output*

Pesan : $22476AA7402D162659C6ED962DBE228BEB124440$

Maka didapkan hasil pembentukan pungsi Has dari sampel dokumen dengan menerapkan algoritma SHA-1 yaitu: $22476AA7402D162659C6ED962DBE228BEB124440$

4. KESIMPULAN

Perdasarkan hasil pembahasan, maka dapat diambil kesimpulan dimana pencarian *file* dokumen yang ganda atau duplikat di dalam sebuah media penyimpanan dapat di lakukan dengan memberikan identitas dari setiap *file* dokumen menggunakan fungsi *hash* sehingga tidak perlu membuka dan membaca isi *file* dokumen satu persatu . Dimana algoritma fungsi *hash* SHA-1 dapat digunakan untuk merepresentasikan isi dari *file* dokumen sehingga dapat digunakan untuk membandingkan *file* dokumen yang ganda atau duplikat.

REFERENCES

[1] R. A. Mollin, An Introduction to Cryptography Second Edition, Florida: Chapman & Hall/CRC, 2007.
[2] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C. Second Edition, New York: John Wiley & Sons, 1996.
[3] F. SUDIRMAN, "Analisis dan Perancangan Sistem Autentikasi Pengguna," Universitas Sumatera Utara, Medan, 2013.
[4] A. Dalimunthe, "Otentikasi Pesan Menggunakan Elliptical Curve Digital Signature Algorithm," Universitas Sumatera Utara, Medan, 2013.
[5] Y. T. P. K. Aryasa, "Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java," Creative Information Technology Journal, vol. 1, no. 1, p. 57, 2015.
[6] Andrus, "Implementasi Modifikasi Sistem Kriptografi RSA dan Elliptic Curve," Universitas Sumatera Utara, Medan, 2016.
[7] S. Patil, N. Jagtap, S. Rajput and R. Sangore, "A Duplicate File Finder System," International Journal of Science Spirituality Business and Technology, pp. 10-14, 2017.
[8] J. Enterprise, Rahasia Manajemen File, Jakarta: Elex Media Komputindo, 2013.