Pelita Informatika : Informasi dan Informatika, Volume 10, Nomor 1, Juli 2021
ISSN 2301-9425 (Media Cetak)
Hal: 12-23

Perancangan Aplikasi Keamanan Duplicate Document Scanner Menerapkan Algoritma SHA-512

Marlina Wahyuni[1]

[1]Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: [1]wahyunimarlina2@gmail.com

Abstrak

Sistem keamanan pengiriman data (komunikasi data yang aman)dipasang untuk mencegah pencurian, kerusakan, dan penyalahgunaan data yang terkirim melalui jaringan komputer. Dalam praktek, pencurian data berwujud pembacaan oleh pihak yang tidak berwenang biasanya dengan menyadap saluran publik. Teknologi jaringan komputer telah dapat mengurangi bahkan membuang kemungkinan adanya kerusakan data akiba tburuknya konektivitas fisik, namun gangguan tetap bisa terjadi karena ada unsur kesengajaan yang mengarah kepenyalahgunaan sistem dari pihak -pihak tertentu. Cryptosystem ini merupakan algoritma yang menyediakan keamanan cukup tinggi yang tidak didasarkan atas kerahasiaan algoritmanya (algoritma restricted), akan tetapi lebih ditekankan pada keamanan/kerahasian kunci yang digunakan(algoritma kriptografi modern). Algoritma restricted biasanya digunakan oleh sekelompok orang untuk bertukar pesan satu samalain, mereka membuat suatu algoritma enkripsi yang hanya diketahui oleh anggota kelompok itu saja, sehingga setiap kali ada anggota kelompok yang keluar, maka algoritma restricted tersebut harus diganti karena kemungkinan anggota kelompok yang keluar itu dapat membocorkan algoritmanya.

Kata Kunci : Kriptografi, Duplicate, Dokumen, Metode SHA-512

1. PENDAHULUAN

Saat ini, teknologi komunikasi dan informasi berkembang dengan pesat dan memberikan pengaruh besar bagi kehidupan manusia. Contoh dari perkembangan ini adalah jaringan *internet*, yang pada saat ini telah memungkinkan banyak orang untuk saling bertukar data secara bebas melalui jaringan tersebut. Karena kemudahan yang dimilikinya, *internet* sudah berkembang menjadi salah satu media yang paling populer di dunia. Namun, kemudahan ini juga dimanfaatkan oleh sebagian pihak yang mencoba untuk melakukan kejahatan. Dengan berbagai teknik, banyak yang mencoba untuk mengakses informasi yang bukan haknya. Oleh karena itu, sejalan dengan berkembangnya media *interne*t ini harus juga dibarengi dengan perkembangan sisi keamanan.

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui oleh pihak yang berhak saja. Apabila informasi tersebut diketahui oleh umum atau pihak lain maka dapat digunakan untuk mendapatkan keuntungan dan dapat digunakan untuk merugikan orang tersebut. *Duplicate* document adalah proses, cara, perbuatan menggandakan. Jadi kata menggandakan dapat diartikan, usaha memperbanyak atau melipatkan beberapa kali dokumen. Dapat diartikan pula penggandaan dokumen, berarti suatu perbuatan. Sistem kriptografi kunci publik memiliki kunci untuk enkripsi K_e dan kunci untuk dekripsi K_d yang berbeda. Kunci untuk enkripsi K_e disebut juga sebagai kunci publik yang bersifat tidak rahasia sehingga dapat didistribusikan melalui saluran tidak aman. Sedangkan kunci dekripsi K_d disebut juga kunci private yang bersifat rahasia dan harus dijaga kerahasiaanya oleh pemegang kunci.

Keamanan data telah menjadi kebutuhan pokok di hampir setiap organissai/perusahaan. Untuk menunjang bisnisnya organisasi/ perusahaan umumnya memerlukan komunikasi. Salah satu kebutuhan keamanan data adalah otentikasi atau jaminan keaslian data. Keaslian data adalah kepastian bahwa data yang ditransmisikan melalui jaringan komputer yang diterima oleh pihak penerima adalah benar data yang dikirimkan oleh pihak pengirim yang dikehendaki pihak penerima. Untuk menjaga keaslian dari *Duplicate Document Scanner* maka digunakan teknik kriptografi dengan menggunakan algoritma SHA 512.

Algoritma SHA-512 jika digunakan untuk menjamin integritas dan keotentikan pesan, dan melakukan kajian skema yang cocok untuk menjaga integritas dan keotentikan pesan yang ditransmisikan di dalam lingkungan intranet. Permasalahan yang akan dipecahkan di dalam penelitian ini adalah bagaimana kekuatan SHA-512 cukup kuat untuk digunakan di dalam menjamin integritas dan keotentikan pesan dan bagaimana skema keamanan yang sesuai diterapkan dengan karakteristik intranet.

Menurut peneliti terdahulu (Hermansyah Sembiring, Fuzy Yustika Manik, Tengku zaidah, 2019, Penerapan Algoritma Secure Hash Algorithm (SHA) Keamanan Pada Citra, MEANS (Media Informasi Analisa dan Sistem),Vol. 4 No. 1), menyatakan bahwa Kriptografi memegang peran penting dalam membangun keamanan citra. Kriptografi bertujuan agar citra tidak dapat dilihat oleh orang yang tidak berhak sehingga informasi baik yang disimpan dalam komputer aman maupun yang dikirim melalui koneksi internet. Serta dapat melindungi kerahasian citra dari berbagai ancaman yang muncul. Algoritma kriptografi yang dapat diterapkan untuk mengamankan citra adalah algoritma Secure Hash Algorithm (SHA) [1].

Menurut peneliti terdahulu (Megah Mulya, Penggunaan Algoritma Sha-512 Untuk Menjamin Integritas Dan Keotentikan Pesan Pada Intranet,Konferensi Nasional Sistem dan Informatika, 2009) bahwa kajian kekuatan algoritma SHA-512 jika digunakan untuk menjamin integritas dan keotentikan pesan, dan melakukan kajian skema yang cocok untuk menjaga integritas dan keotentikan pesan yang ditransmisikan di dalam lingkungan intranet. Permasalahan yang akan dipecahkan di dalam penelitian ini

adalah bagaimana kekuatan SHA-512 cukup kuat untuk digunakan di dalam menjamin integritas dan keotentikan pesan dan bagaimana skema keamanan yang sesuai diterapkan dengan karakteristik intranet [2].

2. TEORITIS

A. Kriptografi

Kriptografi adalah studi yang bertujuan untuk mengamankan dan merahasiakan dengan melakukan proses enkripsi dan dekripsi pada data yang akan diamankan. *Enkripsi* merupakan proses pengubahan data menjadi bentuk sandi yang tidak dipahami dan dibaca, sedangkan dekripsi merupakan proses pengembalian data dalam bentuk sandi ke dalam bentuk semula yang dapat dipahami dan memiliki makna. Dalam kriptografi terdapat beberapa teknik penyandian data yaitu simetris dan asimetris. Kriptografi simetris menggunakan kunci yang sama (kunci simetris) untuk melakukan proses *enkripsi* dan *dekripsi*. Kriptografi asimetris menggunakan kunci yang berbeda untuk proses enkripsi (menggunakan kunci publik) dan *dekripsi* (menggunakan kunci *private*) [8]. Keamanan untuk mengakses dan menyediakan informasi secara cepat dan akurat menjadi sangat esensial bagi sebuah organisasi, baik yang berupa organisasi komersial (perusahaan), perguruan tinggi, lembaga pemerintahan maupun individual (pribadi). Masalah keamanan dan kerahasiaan dan merupakan salah satu aspek penting dari suatu informasi. Dalam hal ini sangat terkait dengan betapa pentingnya informasi akan tidak berguna lagi apabila ditengah jalan informasi itu disadap atau dibajak oleh orang yang tidak berhak.

B. Algoritma SHA-512

SHA adalah fungsi hash satu arah yang didesain oleh National Security Agency (NSA) dan dipublikasi oleh National Institute of Standards and Technology (NIST) sebagai Federal Information Processing Standard (FIPS) pada tahun 1993 dan disebut sebagai SHA-0, dua tahun kemudian dipublikasikan SHA-1 generasi selanjutnya yang merupakan perbaikan dari algoritma SHA-0. Pada tahun 2002 dipublikasikan empat variasi lainnya, yaitu SHA-224, SHA-256, SHA-384, dan SHA-512, keempatnya disebut sebagai SHA-2. SHA dinyatakan aman karena secara komputasi tidak dapat ditemukan isi pesan dari message digest yang dihasilkan, dan tidak dapat dihasilkan dua pesan yang berbeda menghasilkan message digest yang sama. Setiap perubahan yang terjadi pada pesan akan menghasilkan message digest yang berbeda. SHA-512 menggunakan 80 konstanta 64 bit yang sama, yang ditampung pada variable $K0^{(512)}$, $K1^{(512)}$,…$K79^{(512)}$. Konstanta dihasilkan dari proses fractional parts dari *cube roots* pada 80 bilangan prima pertama [11].

Algoritma SHA-512 dirancang oleh *National Security Agency (NSA)*dan dijadikan standard FIPS *(Federal Information Processing Standard)*pada tahun 1993. SHA dibuat berdasarkan fungsi hash MD4 dan desain modelnya menyerupai MD4. Pada tahun 2005, NIST memberikan pengumuman dengan tujuan menghapus setahap demi setahap penggunaan SHA-1 dan mengharapkan pindah pada SHA-2 pada tahun 2010 [2]. Algoritma SHA-512 adalah algoritma yang menggunakan fungsi *hash* satu arah yang diciptakan oleh Ron Rivest. Agoritma merupakan pengembangan dari algoritma-algoritma sebelumnya yaitu algoritma SHA-0, SHA-1, SHA-256 dan agoritma SHA-384.

SHA-512 menggunakan urutan yang sama dari delapan puluh kata konstan 64-bit, K0 {512}, K1{512}, K, K79 {512}. Kata- kata ini mewakili enam puluh empat bit, pertama dari bagian fraksional dari akar kubus dari delapan puluh bilangan prima pertama. Dalam hex, kata-kata konstan ini adalah (dari kiri ke kanan)

428a2f98d728ae22	7137449123ef65cd	b5c0fbcfec4d3b2f	e9b5dba58189dbbc
3956c25bf348b538	59f111f1b605d019	923f82a4af194f9b	ab1c5ed5da6d8118
d807aa98a3030242	12835b0145706fbe	243185be4ee4b28c	550c7dc3d5ffb4e2
72be5d74f27b896f	80deb1fe3b1696b1	9bdc06a725c71235	c19bf174cf692694
e49b69c19ef14ad2	efbe4786384f25e3	0fc19dc68b8cd5b5	240ca1cc77ac9c65
2de92c6f592b0275	4a7484aa6ea6e483	5cb0a9dcbd41fbd4	76f988da831153b5
983e5152ee66dfab	a831c66d2db43210	b00327c898fb213f	bf597fc7beef0ee4
c6e00bf33da88fc2	d5a79147930aa725	06ca6351e003826f	142929670a0e6e70
27b70a8546d22ffc	2e1b21385c26c926	4d2c6dfc5ac42aed	53380d139d95b3df
650a73548baf63de	766a0abb3c77b2a8	81c2c92e47edaee6	92722c851482353b
a2bfe8a14cf10364	a81a664bbc423001	c24b8b70d0f89791	c76c51a30654be30
d192e819d6ef5218	d69906245565a910	f40e35855771202a	106aa07032bbd1b8
19a4c116b8d2d0c8	1e376c085141ab53	2748774cdf8eeb99	34b0bcb5e19b48a8
391c0cb3c5c95a63	4ed8aa4ae3418acb	5b9cca4f7763e373	682e6ff3d6b2b8a3
748f82ee5defb2fc	78a5636f43172f60	84c87814a1f0ab72	8cc702081a6439ec
90befffa23631e28	a4506cebde82bde9	bef9a3f7b2c67915	c67178f2e372532b
ca273eceea26619c	d186b8c721c0c207	eada7dd6cde0eb1e	f57d4f7fee6ed178
06f067aa72176fba	0a637dc5a2c898a6	113f9804bef90dae	1b710b35131c471b
28db77f523047d84	32caab7b40c72493	3c9ebe0a15c9bebc	431d67c49c100d4c
4cc5d4becb3e42b6	597f299cfc657e2a	5fcb6fab3ad6faec	6c44198c4a475817

C. Fungsi Hashing

Fungsi *hashing* (H) merupakan bagian dari algoritma kriptografi yang terdiri dari dua proses utama yaitu enkripsi dan dekripsi. Fungsi Hashing bagian dari fungsi kriptografi modern yang digunakan untuk proses penyandian sebuah dokumen dengan cara meringkas isi dokumen untuk

menghasilkan nilai tertentu (*hash-value*) sehingga dapat berfungsi sebagai identitas pada dokumen tersebut. *Hashvalue* akan berfungsi sebagai sidik jari dokumen digital, kemudian dikombinasikan dengan *Public-key Pair* dan disisipkan ke dalam dokumen digital.

Metode Hashing yang dapat digunakan sebagai fungsi SHA2, yaitu terdiri dari SHA-224, SHA-256, SHA-384 dan SHA-512. Dari keempat fungsi SHA2, maka SHA-256 dan SHA-512 adalah fungsi yang paling umum digunakan untuk meringkas dokumen digital. Pemilihan fungsi SHA2 dilakukan berdasarkan spesifikasi perangkat keras yang akan digunakan dan banyaknya memori yang dibutuhkan untuk dapat menjalankanya. SHA-256 dijalankan pada perangkat dengan processor 32-bit sedangkan SHA-512 dijalankan pada processor 64-bit. Sifat-sifat fungsi *hash* satu arah diantaranya sebagai berikut [2]:

1. Fungsi H dapat diterapkan pada blok data dengan ukuran berapa saja.
2. H menghasilkan nilai (h) dengan panjang tetap (*fixed-lenghth output*).
3. H(x) mudah dihitung untuk setiap nilai x yang diberikan.
4. Untuk setiap h yang diberikan, tidak mungkin menemukan sedemikian sehingga H(x) = h. Itu sebabnya fungsi H dikatakan fungsi *hash* satu-arah (one-way hash function).
5. Untuk setiap x yang diberikan, tidak mungkin mencari pasangan $y \neq x$ sedemikian sehingga H(y) = H(x).
6. Tidak mungkin secara komputasi mencari pasangan x dan y sedemikian sehingga H(x) = H(y).

Sifat-sifat di atas sangat penting untuk sebuah fungsi *hash*, sebab sebuah fungsi *hash* seharusnya berlaku seperti fungsi acak. Sebab fungsi hash dianggap tidak aman jika (i) secara komputasi dimungkinkan menemukan pesan yang bersesuaian dengan pesan ringkasnya (*message digest*), dan (ii) terjadi kolisi (*collision*), yaitu terdapat beberapa pesan berbeda yang mempunyai pesan ringkas yang sama.

3. ANALISA

A. Analisa Masalah

Kegiatan mengamati dan meneliti objek masalah dan menyelesaikan objek masalah tersebut untuk mendapatkan hasil yang sesuai diinginkan berdasarkan ketentuan proses dari pengolahan data yang digunakan. Kegiatan ini meliputi bidang fungsi *Message Diggest*.

Timbulnya kejahatan-kejahatan dalam pengiriman dokumen tentu saja memicu para pakar teknologi informasi untuk meningkatkan keamanan dalam pertukaran informasi. Beberapa penerapan sistem kriptografi sebagai sistem pengaman data antara lain, meningkatkan keamanan informasi dalam sistem berbasis pada kriptografi kunci publik. Menerapkan sistem enkripsi yang dilakukan untuk mencegah terjadinya penyadapan dan kecurangan dan meningkatkan algoritma keamanan. Dalam pengamanan document ini dilakukan pembentukan kunci sehingga menghasilkan nilai, setelah mendapatkan kunci sistem akan mengenkripsi. Setelah mendapatkan pesan terenkripsi selanjutnya dilakukan proses dekripsi dengan menggunakan metode SHA-512

B. Penerapan Metode SHA 512 (*Secure Hash* Algorithm)

Algoritma kriptografi yang digunakan untuk pengamanan dokumen *scanner* mengoptimasi algoritma-algoritma sebelumnya salah satunya adalah SHA 512, yang menjadi objek input yang digunakan dalam penelitian ini adalah dokumen *scanner*. Untuk memudahkan proses analisa maka diambil sampel dari image tersebut berukuran 6 x 6 piksel menggunakan aplikasi Matlab.

Tabel 1. Nilai Desimal

55	53	163	223	214	220
49	54	62	145	217	220
83	63	55	71	220	222
108	78	47	58	121	221
101	112	55	55	89	218
108	106	66	49	78	211

Data decimal dikonversi menjadi *binary*. Berikut hasil konversi dalam *biner* pada tabel 2.

Tabel 2. Nilai *Input* biner

00110111	00110101	10100011	11011111	11010110	11011100
00110001	00110110	00111110	10010001	11011001	11011100
01010011	00110101	00110111	01000111	11011100	11011110
01101100	01001110	00101111	00111010	01111001	11011101
01100101	01110000	00110111	00110111	01011001	11011010
01101100	01101010	01000010	00110001	01001110	11010011

1. Penambahan *Padding Bit*

   Dari tabel di atas diketahui bahwa panjang M=288 bit. Proses berikutnya adalah dengan menambahkan *padding* bit 1 dan sisanya 0 sejumlah *k*, dengan persamaan sebagai berikut :

   $l + 1 + k = 896 \bmod 1024$
   $288 + l + k = 896 \bmod 1024$
   $288 + k = 896 \bmod 1024$
   $k = 896 - 289 \bmod 1024$
   $k = 607$

   Maka banyaknya *padding* bit 0 yang ditambahkan adalah 607 bit.

Tabel 3. Penambahan *Padding* Bit

0011	0011	1010	1101	1101	1101	0011	0011
0111	0101	0011	1111	0110	1100	0001	0110
0011	1001	1101	1101	0101	0011	0011	0100
1110	0001	1001	1100	0011	0101	0111	0111
1101	1101	0110	0100	0010	0011	0111	1101
1100	1110	1100	1110	1111	1010	1001	1101
0110	0111	0011	0011	0101	1101	0110	0110
0101	0000	0111	0111	1001	1010	1100	1010

0100	0011	0100	1101	**1000**	**0000**	**0000**	**0000**
0010	0001	1110	0011	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**

2. Penambahan Panjang *Append*

   Penambahan panjang *append* dilakukan dengan penambahan panjang pesan sebanyak 64 bit di akhir. Panjang pesan adalah 288 bit sehingga ditambahkan panjang *append* sebagai berikut :

Tabel 4. Penambahan Panjang *Append*

00110111	00110101	10100011	11011111	11010110	11011100	00110001	00110110
00111110	10010001	11011001	11011100	01010011	00110101	00110111	01000111
11011100	11011110	01101100	01001110	00101111	00111010	01111001	11011101
01100101	01110000	00110111	00110111	01011001	11011010	01101100	01101010
01000010	00110001	01001110	11010011	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	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**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000001**	**00100000**

3. *Parsing* Pesan

   Pada tahap ini panjang pesan tidak lebih dari 512 sehingga hanya menghasilkan 1 blok 512 bit Yaitu $W^{(0)}$. Tahap selanjutnya adalah melakukan parsing dengan membagi setiap blok 512 bit menjadi 16 blok berukuran 64 bit

Tabel 5. *Parsing* Pesan

W_0	00110111 00110101 10100011 11011111 11010110 11011100 00110001 00110110
W_1	00111110 10010001 11011001 11011100 01010011 00110101 00110111 01000111
W_2	11011100 11011110 01101100 01001110 00101111 00111010 01111001 11011101
W_3	01100101 01110000 00110111 00110111 01011001 11011010 01101100 01101010
W_4	01000010 00110001 01001110 11010011 10000000 00000000 00000000 00000000
W_5	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_6	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_7	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_8	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_9	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{10}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{11}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{12}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{13}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{14}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{15}	00000000 00000000 00000000 00000000 00000000 00000000 00000001 00100000

4. Proses Enkripsi

   Kemudian dilakukan proses Enkripsi, langkah ini dilakukan dengan ketentuan sebagai berikut :

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ s_1^{\{512\}}(Wt\text{-}2) + W\,t\text{-}7 + s\,0\{512\}(Wt\text{-}15) + Wt\text{-}16 & 16 \le t \le 79 \end{cases}$$

Wt = Blok pesan yang baru
Mt = Blok pesan yang lama
W_{i-2} = Blok pesan dari W ke i-2
W_{i-15} = Blok pesan dari W ke i-15
ROTR = RotateRight
SHR = ShiftRight
$\oplus$ = Operator XOR

Untuk proses ke 16 sampai 79 dilakukan perhitungan sebagai berikut :

Putaran 1:

$W_{16} = \partial_1^{(512)}(W_{t-2}) + W\ t\ \text{-}7 + \partial_0^{\{512\}}(W_{t-15}) + W_{t-16}$

$W_{16} = \partial_1^{(512)}(W_{16-2}) + W_{16}\text{-}_7 + \partial_0^{\{512\}}(W_{16-15}) + W_{16-16}$

$W_{16} = \partial_1^{(512)}(W_{14}) + W_9 + \partial_0^{\{512\}}(W_1) + W_0$

$\partial_1^{(512)}(W_{14}) = ROTR^{19}(W_{14}) \oplus ROTR^{61}(W_{14}) \oplus SHR^6(W_{14})$

$ROTR^{19}(W_{14})$=**0000000000000000000**0000000000000000 0000000000000000000000000000

$ROTR^{61}(W_{14})$=**0000000000000000000**0000000000000000 0000000000000000000000000000

$SHR^6(W_{14})$ =
0000000000000000000000000000000000000000000000000 000000000000000

0000000000000000000000000000000000000000000000000 000000000000000

Keterangan

1. Untuk pencarian $ROTR^{19}$ diambil dari nilai W_{14} dengan cara memindahkan sebanyak 19 biner dari kanan ke kiri.
2. Untuk pencarian $ROTR^{61}$ diambil dari nilai W_{14} dengan cara memindahkan sebanyak 61 biner dari kanan ke kiri.
3. Untuk pencarian SHR^{6} diambil dari nilai W_{14} dengan cara menghapus nilai dari kanan sebanyak enam kemudian menuliskan 0 sebanyak enam ke kiri.
4. Hasil dari $ROTR^{19}$, $ROTR^{61}$ dan SHR^{6} Kemudian di Xor kan
5. Untuk mendapatkan hasil W^{16} seluruh hasil W_{14}, W_9, W_1 dan W_0 akan ditambahkan

$\partial_0^{(512)}(W_1) = ROTR^1(W_1) \oplus ROTR^8(W_1) \oplus SHR^7(W_1)$

$ROTR^1(W_1)$ =
100111110100100011101100111011100010100110011010100110111010 0011

$ROTR^8(W_1)$ =
0100011100111110100100011101100111011100010100110011101010011 0111

$SHR^7(W_1)$ =
000000000111110100100011101100111011100010100110011010100110 1110

1101100000001011010111101000010001001101011011111100010011111010

$W_{16} = \partial_1^{(512)}(W_{14}) + W_9 + \partial_0^{\{512\}}(W_1) + W_0$

W_{16} =
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
1101100000001011010111101000010001001101011011111100010011111010
0011011100110101101000111101111111010110110111000011000100110110 +
0000111101000000000000101100011001000110100101111110101100110000

Demikian proses seterusnya hingga $W_{79}^{(i-1)}$, di mana I adalah jumlah blok 512 bit. Hasil seluruh proses tersebut dapat dilihat pada Tabel 6. di bawah ini

Tabel 6. Penjadwalan Pesan

W_0	00110111 00110101 10100011 11011111 11010110 11011100 00110001 00110110
W_1	00111110 10010001 11011001 11011100 01010011 00110101 00110111 01000111
W_2	11011100 11011110 01101100 01001110 00101111 00111010 01111001 11011101
W_3	01100101 01110000 00110111 00110111 01011001 11011010 01101100 01101010
W_4	01000010 00110001 01001110 11010011 10000000 00000000 00000000 00000000
W_5	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_6	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_7	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_8	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_9	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{10}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{11}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{12}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{13}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{14}	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
W_{15}	00000000 00000000 00000000 00000000 00000000 00000000 00000001 00100000
W_{16}	00001111 01000000 00000001 01100011 00100011 01001011 11110101 00110000
W_{17}	01110000 10011011 00101101 01101111 00011000 00100001 10101001 10101011
W_{18}	00110100 01101010 10111011 10011011 01100010 01110100 11110010 00000010
W_{19}	10100010 01111100 10010100 11001111 00100101 00011010 10010100 01100011
W_{20}	01111111 11110101 01111110 10001011 10101001 00000000 00000000 00000000
W_{21}	01000011 11100001 00100000 01100101 10000110 11011001 00101001 11101111
W_{22}	11111110 01010100 00101110 01011001 11001000 01110101 01110110 01000011
W_{23}	01001010 01111011 01110000 00110100 10100111 00101001 10010000 00110110
W_{24}	10011101 00100001 00110110 10011001 11110100 01110110 10010111 01010010
W_{25}	10010100 01011110 01100000 11000101 11100111 01001010 10100101 00011001
W_{26}	11011011 00010011 11110111 01111111 00000111 11010000 11110010 10101010
W_{27}	01110011 11110110 11001011 10110010 10001010 11010000 10111110 10110001
W_{28}	00001000 10000111 11001111 10101100 01000010 01010000 01011111 01011101
W_{29}	10000111 00000011 11010110 00011101 00001101 00111101 01011100 01101110
W_{30}	10111001 01110010 00110011 10000000 00000001 10100111 11000011 10101010
W_{31}	01100101 01000010 11100101 10011110 11011001 10011110 11101101 11001101
W_{32}	11100111 01111011 10011111 11001000 10010010 00010101 11010010 11100111
W_{33}	01011101 01010010 10010001 10110111 00100110 10101111 01001101 01011011
W_{34}	11011100 01010101 01000001 10111110 01110000 01000101 11100111 01110110
W_{35}	11101011 11000111 10101111 10010110 01101010 10111011 00100110 11111010
W_{36}	10110001 01100001 10001010 01111001 11110110 11111010 00111101 00011110
W_{37}	11110010 11001000 01111101 10111000 10001110

	01100011 00100110 01101111
W_{38}	01000100 11100011 01001011 01101011 00010000 11010101 01010011 10011001
W_{39}	00111111 01110010 00111001 01111110 00100111 11111100 11100000 01010111
W_{40}	01011000 10000000 11000101 11111100 00011010 00110101 00011100 10101110
W_{41}	10011101 11111101 01010100 00100011 00011001 11000110 10011001 10011000
W_{42}	00110100 01001101 00001011 10000001 00111100 00000001 11111000 10000101
W_{43}	00111011 01011111 00001011 10101000 00111101 10001111 10001111 10111000
W_{44}	11000011 00000000 00001001 10000110 10110110 11001011 01010010 11101010
W_{45}	11101101 00111010 11010110 00110000 01001010 11100010 11111100 11100110
W_{46}	11101000 01000011 01101000 01010010 10100100 11111001 11100100 00111110
W_{47}	00000111 01010100 01110100 01011011 11001111 10001001 01000101 00010111
W_{48}	10110001 10000110 10110101 01101001 11110110 11011100 11010111 00000111
W_{49}	10111100 00001001 00101110 00111010 11111001 01100001 11110110 11001111
W_{50}	10111001 10100110 00110010 01001011 00000001 10000110 00001001 11111110
W_{51}	01001101 00010100 11000001 10101100 10010110 11111001 11110110 00110110
W_{52}	01000011 11110110 10100011 10101000 10110100 11111100 01001100 01000011
W_{53}	01101100 00011011 11111101 00110010 00000100 01011000 10100100 10011010
W_{54}	10101010 01100001 11110000 00101111 01000001 00111000 10010010 01101111
W_{55}	11100111 10011101 10011100 01111110 10111011 01111110 10001110 10011010
W_{56}	00101110 11001011 10010011 00011001 10000110 00100100 10110010 11110110
W_{57}	11110001 10111110 01001111 11000001 11001001 11100010 11111001 01100011
W_{58}	10000110 11111100 10100001 01001110 11011101 11010101 01111010 11000101
W_{59}	11011001 00110010 10101011 00101111 01000100 00000001 01001000 01101011
W_{60}	01011010 01101011 11011101 01000110 01110000 11000110 10100111 01110111
W_{61}	11000101 10110000 00011111 11100000 11111000 00101010 00110100 10000111
W_{62}	11111011 01111111 10110110 01110111 11100100 11001010 11100101 10001010
W_{63}	01111100 00110110 10011111 10101101 11010111 01101111 00011000 11110001
W_{64}	00110110 10100101 10100010 10100010 00000001 10010011 00100100 10101100
W_{65}	11101000 01111000 11001010 10010001 00010010 11000111 10111110 01111110
W_{66}	01110011 10010111 00111111 01011100 00110011 00011001 01101011 00101011
W_{67}	01000000 01100110 01010010 11001111 01110100 10000110 10001110 11111000
W_{68}	01100100 01110001 01011100 10010100 00001111 00110010 11110101 11111110
W_{69}	10011110 00001100 11111010 00101100 10111011 10111010 00100010 01100111
W_{70}	10001010 00100001 11100101 11010011 11111000 01010001 01010011 00000000
W_{71}	10110100 10101011 11100100 10100101 10010101 00100011 00100110 10110100
W_{72}	00101001 01010111 11111010 00100110 11101111 01100111 01101101 01110000
W_{73}	11101110 10100101 11111111 10001100 11111100 01100000 00110001 00010110
W_{74}	11110011 00001000 10000000 01111010 01000011 11101001 11011010 10000010
W_{75}	00000111 11000101 10011010 01010110 00110010 11111010 01111011 11011111
W_{76}	11111100 11100101 00110101 11110100 01101110 11101011 01101000 11111100
W_{77}	10110110 11001011 10000011 10110100 11010110 10010101 10000101 10111010
W_{78}	10000111 10001110 01010010 01111010 10010110 00011001 10101111 00101000
W_{79}	01100000 01010110 00111101 11111010 11000011 00100001 10001101 10101101

5. Inisialisasi Nilai *Hash*
   Dimana nilai *Hash* ini merupakan sebuah ketentuan yaitu
   $H_0^{(0)}$ = 6a09e667f3bcc908
   $H_1^{(0)}$ = bb67ae8584caa73b
   $H_2^{(0)}$ = 3c6ef372fe94f82b
   $H_3^{(0)}$ = a54ff53a5f1d36f1
   $H_4^{(0)}$ = 510e527fade682d1
   $H_5^{(0)}$ = 9b05688c2b3e6c1f
   $H_6^{(0)}$ = 1f83d9abfb41bd6b
   $H_7^{(0)}$ = 5be0cd19137e2179
6. Proses Dekripsi
   Selanjutnya melakukan kompresi a, b, c, d, e, f, g, dan h di mana setiap variabel diambil dari *initial Hash value*
   $a=H_0^{(0)}$,
   $b=H_1^{(0)}$,
   $c=H_2^{(0)}$,
   $d=H_3^{(0)}$,
   $e=H_4^{(0)}$,
   $f=H_5^{(0)}$,
   $g=H_6^{(0)}$,
   $h=H_7^{(0)}$.

Selanjutnya dilakukan proses komputasi fungsi *Hash* SHA-512 dari t=0 sampai t=79. Untuk mencari nilai T_0 dilakukan perhitungan melalai nilai Init atau nilai ketentuan awal dengan perhitungan sebagai berikut :

$T_1 = h + \sum_1^{\{512\}} (e) + Ch(e, f , g) + K_t^{\{512\}} + W_t$

$T_2 = \sum_0^{\{512\}}(a) + Maj(a, b,c)$

Dimana nilai a, b, c, d, e, f, g, dan h berikutnya diambil dengan ketentuan

$a = T_1 + T_2$, $b = a$, $c = b$, $d = c$, $e = d + T_1$, $f = e$, $g = f$, $h = g$.

Putaran 1:

$T_1 = h + \sum_1^{\{512\}}(e) + Ch(e, f, g) + K_0^{\{512\}} + W_0$

$T_1 = 5BE0CD19137E2179 + \sum_1^{\{512\}}(510E527FADE682D1) + Ch(e, f, g) + 428A2F98D728AE22 + 3735A3DFD6DC3136$

$\sum_1^{\{512\}}(e) = ROTR^{14}(e) \oplus ROTR^{18}(e) \oplus SHR^{41}(e)$

$\sum_1^{\{512\}}(510E527FADE682D1)$

$= A0B45443949FEB79 + A0B45443949FEB79 + 3FD6F34168A88729$

$= 813F9BC891E85E1B$

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

$Ch(510E527FADE682D1, 9B05688C2B3E6C1F, 1F83D9ABFB41BD6B)$

$= (510E527FADE682D1 \wedge 9B05688C2B3E6C1F) \oplus (510E527FADE682D1 \wedge 1F83D9ABFB41BD6B)$

$= 1104400C29260011 \oplus 1102502BA9408041$

$= 0006102780668050$

$T_1 = h + \sum_1^{\{512\}}(e) + Ch(e, f, g) + K_0^{\{512\}} + W_0$

$T_1 = 5BE0CD19137E2179 + 813F9BC891E85E1B + 0006102780668050 + 428A2F98D728AE22 + 3735A3DFD6DC3136$

$T_1 = 56E64C81D3D1DF3C$

$T_2 = \sum_0^{\{512\}}(a) + Maj(a, b, c)$

$= \sum_1^{\{512\}}(a) = ROTR^{28}(a) \oplus ROTR^{34}(a) \oplus SHR^{39}(a)$

$= \sum_0^{\{512\}}(6A09E667F3BCC908)$

$= 3BCC9086A09E667F \oplus FCEF32421A827999 \oplus CFE7799210D413CC$

$= 8C4DB56AAC80C2A$

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

$Maj(6A09E667F3BCC908, BB67AE8584CAA73B, 3C6EF372FE94F82B)$

$= (6A09E667F3BCC908 \wedge BB67AE8584CAA73B) \oplus (6A09E667F3BCC908 \wedge 3C6EF372FE94F82B) \oplus (BB67AE8584CAA73B \wedge 3C6EF372FE94F82B)$

$= 2A01A60580888108 \oplus 2808E262F294C808 \oplus 3866A2008480A02B$

$= 3A6FE667F69CE92B$

$T_2 = \sum_0^{\{512\}}(a) + Maj(a, b, c)$

$T_2 = 8C4DB56AAC80C2A + 3A6FE667F69CE92B$

$= 4334C1BEA164F555$

Demikian proses komputasi seterusnya hingga mencari nilai t=79. Hasil seluruh proses tersebut dapat dilihat pada Tabel 7. di bawah ini

Tabel 7. Proses Komputasi Fungsi *Hash* SHA 512

	A	B	C	D	E	F	G	H
In	6A09E667F3BC C908	BB67AE85 84CA A73B	3C6E F372 FE94 F82B	A54F F53A 5F1D 36F1	510E 527F ADE 682D 1	9B05 688C 2B3E 6C1F	1F83 D9A BFB4 1BD6 B	5BE0 CD19 137E 2179
T0	613D AC7 BDB 42DC 33	6A09 E667 F3BC C908	BB67 AE85 84CA A73B	3C6E F372 FE94 F82B	FC36 41BC 32EF 162D	510E 527F ADE 682D 1	9B05 688C 2B3E 6C1F	1F83 D9A BFB4 1BD6 B
T1	C931 78D5 4EE3	613D AC7 BDB	6A09 E667 F3BC	BB67 AE85 84CA	DA74 F128 F590	FC36 41BC 32EF	510E 527F ADE	9B05 688C 2B3E
T2	3629 DE27 4672 AF9E	C931 78D5 4EE3	613D AC7 BDB	6A09 E667 F3BC	713C 9030 6090 DD16	DA74 F128 F590	FC36 41BC 32EF	510e5 27fad e682d 1
T3	F510 21EB AB0 B303 2	3629 DE27 4672 AF9E	C931 78D5 4EE3 E2F7	613D AC7 BDB 42DC 33	856A ECA 8AD0 CA04 4	713C 9030 6090 DD16	DA74 F128 F590 3CB2	FC36 41BC 32EF 162D
T4	7D0E 3D70 5056 3A3F	F510 21EB AB0 B303 2	3629 DE27 4672 AF9E	C931 78D5 4EE3 E2F7	5BF2 97C1 5389 3D39	856A ECA 8AD0 CA04 4	713C 9030 6090 DD16	DA74 F128 F590 3CB2
T5	CDA E932 8A62 3ABC9	7D0E 3D70 5056 3A3F	F510 21EB AB0 B303 2	3629 DE27 4672 AF9E	A8E C8E1 8A25 B105 F	5BF2 97C1 5389 3D39	856A ECA 8AD0 CA04 4	713C 9030 6090 DD16
T6	84FB 874B 4CD D6D 68	CDA E932 8A62 3ABC9	7D0E 3D70 5056 AB0 B303 2	F510 21EB AB0 B303 2	9421 27D9 18E4 217B	A8E C8E1 8A25 B105 F	5BF2 97C1 5389 3D39	856A ECA 8AD0 CA04 4
T7	0E35 36D1 819B 1954	84FB 874B 4CD D6D 68	CDA E932 8A62 3ABC9	7D0E 3D70 5056 3A3F	B80E 6860 7DB4 8B83	9421 27D9 18E4 217B	A8E C8E1 8A25 B105 F	5BF2 97C1 5389 3D39
T8	F040 52ED 6F537	0E35 36D1 819B 1954	84FB 874B 4CD D6D 68	CDA E932 8A62 3ABC9	D2E1 B2C1 A07A BB6D	B80E 6860 7DB4 8B83	9421 27D9 18E4 217B	A8E C8E1 8A25 B105 F
T9	3A3B 943E 21DC 3A7E	F040 52ED 6F01 F537	0E35 36D1 819B 1954	84FB 874B 4CD D6D 68	1277 E2C4 E270 AE0B	D2E1 B2C1 A07A BB6D	B80E 6860 7DB4 8B83	9421 27D9 18E4 217B
T10	BF14 2276 879D 472D	3A3B 943E 21DC 3A7E	F040 52ED 6F01 F537	0E35 36D1 819B 1954	1FE7 81BB E19D A95A	1277 E2C4 E270 AE0B	D2E1 B2C1 A07A BB6D	B80E 6860 7DB4 8B83
T11	DEE 050F 045C 64C12	BF14 2276 879D 472D	3A3B 943E 21DC 3A7E	F040 52ED 6F01 F537	FDC9 5BA4 D9C AE676	1FE7 81BB E19D A95A	1277 E2C4 E270 AE0B	D2E1 B2C1 A07A BB6D
T12	9368 0FF4 FD24 566A	DEE 050F 045C 64C12	BF14 2276 879D 472D	3A3B 943E 21DC 3A7E	17D9 73BA EA49 21CA	FDC9 5BA4 D9C AE676	1FE7 81BB E19D A95A	1277 E2C4 E270 AE0B
T13	E66D 71FD B5C1 5954	9368 0FF4 FD24 566A	DEE 050F 045C 64C12	BF14 2276 879D 472D	D3A A249 5961 8720F	17D9 73BA EA49 21CA	FDC9 5BA4 D9C AE676	1FE7 81BB E19D A95A
T14	9394 495E EEF9 5363	E66D 71FD B5C1 5954	9368 0FF4 FD24 566A	DEE 050F 045C 64C12	B231 D836 8A51 A4F7	D3A A249 5961 8720F	17D9 73BA EA49 21CA	FDC9 5BA4 D9C AE676
T15	5DE2 44A6 A2A	9394 495E EEF9	E66D 71FD B5C1	9368 0FF4 FD24	BF5F BB52 D007	B231 D836 8A51	D3A A249 5961	17D9 73BA EA49

	A	B	C	D	E	F	G	H
	914D7	5363	5954	566A	5E5C	A4F7	8720F	21CA
T16	6901DF118985DDD5	5DE244A6A2A914D7	9394495EEEF95363	E66D71FDB5C15954	7F0714C26E4231B2	BF5FBB52D0075E5C	B231D8368A51A4F7	D3AA2495961 8720F
T17	7129752E2224EF3B	6901DF118985DDD5	5DE244A6A2A914D7	9394495EEEF95363	24E6F87CCCAD6D56	7F0714C26E4231B2	BF5FBB52D0075E5C	B231D8368A51A4F7
T18	7F123980 6E3A8AE7	7129752E2224EF3B	6901DF118985DDD5	5DE244A6A2A914D7	2F783B17FBB6	24E6F87CCCAD6D56	7F0714C26E4231B2	BF5FBB52D0075E5C
T19	6AE7A34F23E1DB16	7F123980 6E3A8AE7	7129752E2224EF3B	6901DF118985DDD5	56CDF119BC4CED E1	2F783B17FBB51556	24E6F87CCCAD6D56	7F0714C26E4231B2
T20	F5335033599A46DD	6AE7A34F23E1DB16	7F123980 6E3A8AE7	7129752E2224EF3B	7A6FFA21A8E94614	56CDF119BC4CED E1	2F783B17FBB51556	24E6F87CCCAD6D56
T21	06A0021EE99EEE90	F5335033599A46DD	6AE7A34F23E1DB16	7F123980 6E3A8AE7	CF247D640A75615B	7A6FFA21A8E94614	56CDF119BC4CED E1	2F783B17FBB51556
T22	787F7604CA18B8C7	06A0021EE99EEE90	F5335033599A46DD	6AE7A34F23E1DB16	8F09066DD51313AD	CF247D640A75615B	7A6FFA21A8E94614	56CDF119BC4CED E1
T23	3C4BA7FC81CC3D28	787F7604CA18B8C7	06A0021EE99EEE90	F5335033599A46DD	986665FD04FAE648	8F09066DD51313AD	CF247D640A75615B	7A6FFA21A8E94614
T24	52987694 89CB9D61	3C4BA7FC81CC3D28	787F7604CA18B8C7	06A0021EE99EEE90	47CBC6C7E365E43E	986665FD04FAE648	8F09066DD51313AD	CF247D640A75615B
T25	D0DA9CE74E4C14FA	52987694 89CB9D61	3C4BA7FC81CC3D28	787F7604CA18B8C7	0F32545D335A0741	47CBC6C7E365E43E	9866 65FD04FAE648	8F09066DD51313AD
	A	B	C	D	E	F	G	H
T26	0E72AB0D2953FAB2	D0DA9CE74E4C14FA	52987694 89CB9D61	3C4BA7FC81CC3D28	3C489763 1D61B99D	0F32545D335A0741	47CBC6C7E365E43E	9866 65FD04FAE648
T27	6A2F650E9404384F	0E72AB0D2953FAB2	D0DA9CE74E4C14FA	52987694 89CB9D61	474CAF5E9199D93E	3C489763 1D61B99D	0F32545D335A0741	47CBC6C7E365E43E
T28	C6AFEE5E23E E105A	6A2F650E9404384F	0E72AB0D2953FAB2	D0DA9CE74E4C14FA	52987694 89CB9D61	474CAF5E9199D93E	3C489763 1D61B99D	0F32545D335A0741
T	D8B7	C6A	6A2F	0E72	5104	5298	474C	3C48
	2208	FEE5	650E	AB0	7247	7694	AF5E	9763
29	8FCB361E	E23E E105A	9404384F	D2953FAB2	4DF8D115	89CB9D61	9199D93E	1D61B99D
T30	6538A91FECEDF043	D8B72208 8FCB361E	C6AFEE5 E23E E105A	6A2F650E 9404384F	DA9DB41 88D85344 9	5104 7247 4DF8D115	5298 7694 89CB9D61	474CAF5E 9199D93E
T31	E4445685A91F ECED7C	6538A91F ECEBC96 4	D8B72208 8FCB361E	C6AFEE5 E23E E105A	7F76 6DA FAF D2066E	DA9DB41 88D85344 9	5104 7247 4DF8D115	5298 7694 89CB9D61
T32	82FA5E591B88CC24	E4445685 D7CBC96 4	6538A91F ECEDF043	D8B72208 8FCB361E	6294 3723 BB0DF6E7	7F76 6DA FAF D2066E	DA9DB41 88D85344 9	5104 7247 4DF8D115
T33	4BC C518E699B571 5	82FA5E591B88CC24	E4445685 D7CBC96 4	6538A91F ECEDF043	F3B1349F D37E3E14	6294 3723 BB0DF6E7	7F76 6DA FAF D2066E	DA9DB41 88D85344 9
T34	B549018E83A27BB1	4BC C518E699B571 5	82FA5E591B88CC24	E4445685 D7CBC96 4	8D45F85F EEFE9E1D	F3B1349F D37E3E14	6294 3723 BB0DF6E7	7F76 6DA FAF D2066E
T35	640FEE41AE72BB31	B549018E83A27BB1	4BC C518E699B571 5	82FA5E591B88CC24	3CC39239 200B5AD8	8D45F85F EEFE9E1D	F3B1349F D37E3E14	6294 3723 BB0DF6E7
T36	B462E36914288467	FE2A99ED89B0 1388	C4B B6F667E1 7BB35	4BC C518E699B571 5	C2240787 08C9D504	3CC39239 200B5AD8	8D45F85F EEFE9E1D	F3B1349F D37E3E14
T37	6661403D2664AE2E	B462E369 14288467	FE2A99ED89B0 1388	C4B B6F667E1 7BB35	E858 6FF4 0F712691	C2240787 08C9D504	3CC39239 200B5AD8	8D45F85F EEFE9E1D
T38	9559E257 27FD85DE	6661403D 2664AE2E	B462E369 14288467	FE2A99ED 89B01388	0E13901E FEB42988	E858 6FF4 0F712691	C2240787 08C9D504	3CC39239 200B5AD8
T39	6CB7E275 5FFACE9D	9559E257 27FD85DE	6661403D 2664AE2E	B462E369 14288467	9E17D250 755B6F94	0E13901E FEB42988	E858 6FF4 0F712691	C2240787 08C9D504
T40	B178DE93 89046A42	6CB7E275 5FFACE9D	9559E257 27FD85DE	6661403D 2664AE2E	245C3EC8 67F0F7E1	9E17D250 755B6F94	0E13901E FEB42988	E858 6FF4 0F712691
T41	A9AAAB C69449E183	B178DE93 89046A42	6CB7E275 5FFACE9D	9559E257 27FD85DE	057C533D 6C0091B2	245C3EC8 67F0F7E1	9E17D250 755B6F94	0E13901E FEB42988
T42	F7DB37B 11DBC391E	A9AAAB C694 49E183	B178DE93 89046A42	6CB7E275 5FFACE9D	76D53B97 30DD90AA	057C533D 6C0091B2	245C3EC8 67F0F7E1	9E17D250 755B6F94
T43	3F3F3D72	F7DB37B 11DBC694	A9AAAB C694	B178DE93 8904	FB647878	76D53B97 30DD	057C533D 6C00	245C3EC8 67F0

A	B	C	D	E	F	G	H
0035 FEBF	C391 E	49E1 83	6A42	C57A 63C8	90AA	91B2	F7E1

	A	B	C	D	E	F	G	H
T	6F89	3F3F	F7D	A9A	F9F4	FB64	76D5	057C
4	E9F4	3D72	B37B	AAB	88F4	7878	3B97	533D
4	A60	0035	11DB	C694	D891	C57A	30DD	6C00
	DD9	FEBF	C391	49E1	E431	63C8	90AA	91B2
	94		E	83				
T	F729	6F89	3F3F	F7D	8125	F9F4	FB64	76D5
4	7FD3	E9F4	3D72	B37B	F919	88F4	7878	3B97
5	76AE	A60	0035	11DB	3B16	D891	C57A	30DD
	CB91	DD9	FEBF	C391	BC92	E431	63C8	90AA
		94		E				
T	9373	F729	6F89	3F3F	FA7	8125	F9F4	FB64
4	7FEA	7FD3	E9F4	3D72	AE33	F919	88F4	7878
6	EC0	76AE	A60	0035	DB6	3B16	D891	C57A
	D47C	CB91	DD9	FEBF	E7B5	BC92	E431	63C8
	4		94		11			
T	F97B	9373	F729	6F89	38BA	FA7	8125	F9F4
4	38D6	7FEA	7FD3	E9F4	7648	AE33	F919	88F4
7	49B4	EC0	76AE	A60	49EA	DB6	3B16	D891
	ED46	D47C	CB91	DD9	EC05	E7B5	BC92	E431
		4		94		11		
T	06D4	F97B	9373	F729	765E	38BA	FA7	8125
4	DF95	38D6	7FEA	7FD3	C98A	7648	AE33	F919
8	76EF	49B4	EC0	76AE	1CFD	49EA	DB6	3B16
	4C4D	ED46	D47C	CB91	25E1	EC05	E7B5	BC92
			4				11	
T	58E4	06D4	F97B	9373	A2F7	765E	38BA	FA7
4	8C75	DF95	38D6	7FEA	C316	C98A	7648	AE33
9	0BD6	76EF	49B4	EC0	5EE2	1CFD	49EA	DB6
	1619	4C4D	ED46	D47C	2D59	25E1	EC05	E7B5
				4				11
T	F441	58E4	06D4	F97B	193D	A2F7	765E	38BA
5	D12B	8C75	DF95	38D6	6C01	C316	C98A	7648
0	4C46	0BD6	76EF	49B4	019D	5EE2	1CFD	49EA
	AA8	1619	4C4D	ED46	78AC	2D59	25E1	EC05
	A							
T	06F3	F441	58E4	06D4	69C6	193D	A2F7	765E
5	C492	D12B	8C75	DF95	90DA	6C01	C316	C98A
1	2AF8	4C46	0BD6	76EF	3901	019D	5EE2	1CFD
	623F	AA8	1619	4C4D	5C20	78AC	2D59	25E1
		A						
T	CF3B	06F3	F441	58E4	7FC1	69C6	193D	A2F7
5	DEF6	C492	D12B	8C75	2E9C	90DA	6C01	C316
2	3F57	2AF8	4C46	0BD6	DAE	3901	019D	5EE2
	0608	623F	AA8	1619	2D0A	5C20	78AC	2D59
			A		B			
T	6D6F	CF3B	06F3	F441	B633	7FC1	69C6	193D
5	FB33	DEF6	C492	D12B	702A	2E9C	90DA	6C01
3	01C1	3F57	2AF8	4C46	0C19	DAE	3901	019D
	D8C7	0608	623F	AA8	A786	2D0A	5C20	78AC
				A		B		
T	FFA6	6D6F	CF3B	06F3	E9FD	B633	7FC1	69C6
5	6B21	FB33	DEF6	C492	E3A4	702A	2E9C	90DA
4	97EA	01C1	3F57	2AF8	34E1	0C19	DAE	3901
	68F1	D8C7	0608	623F		A786	2D0A	5C20
					3F62		B	
T	1220	FFA6	6D6F	CF3B	D16A	E9FD	B633	7FC1
5	B936	6B21	FB33	DEF6	4BF1	E3A4	702A	2E9C
5	D9F6	97EA	01C1	3F57	5EA8	34E1	0C19	DAE
	7FC2	68F1	D8C7	0608	2F31	3F62	A786	2D0A
								B
T	32D4	1220	FFA6	6D6F	7A53	D16A	E9FD	B633
5	2282	B936	6B21	FB33	D460	4BF1	E3A4	702A
6	6EEF	D9F6	97EA	01C1	563C	5EA8	34E1	0C19
	7C78	7FC2	68F1	D8C7	0D6B	2F31	3F62	A786
T	B70F	32D4	1220	FFA6	247F	7A53	D16A	E9FD
5	D74C	2282	B936	6B21	D27F	D460	4BF1	E3A4
7	269C	6EEF	D9F6	97EA	285E	563C	5EA8	34E1
	C417	7C78	7FC2	68F1	9CD	0D6B	2F31	3F62
					E			
T	BBD	B70F	32D4	1220	9159	247F	7A53	D16A
5	D99B	D74C	2282	B936	07D6	D27F	D460	4BF1

	A	B	C	D	E	F	G	H
8	1B28	269C	6EEF	D9F6	2926	285E	563C	5EA8
	AF23	C417	7C78	7FC2	4014	9CD	0D6B	2F31
	4					E		
T	89D4	BBD	B70F	32D4	C0AF	9159	247F	7A53
5	705B	D99B	D74C	2282	0BEF	07D6	D27F	D460
9	83B0	1B28	269C	6EEF	3BE	2926	285E	563C
	63DB	AF23	C417	7C78	A925	4014	9CD	0D6B
		4					E	
T	D2E	89D4	BBD	B70F	05C2	C0AF	9159	247F
6	E592	705B	D99B	D74C	7BA	0BEF	07D6	D27F
0	7E8D	83B0	1B28	269C	A57C	3CBE	2926	285E
	352A	63DB	AF23	C417	2CF1	A925	4014	9CD
	4		4		C			E
T	9E99	D2E	89D4	BBD	55A9	05C2	C0AF	9159
6	D146	E592	705B	D99B	A892	7BA	0BEF	07D6
1	BCB	7E8D	83B0	1B28	E350	A57C	3CBE	2926
	3DC	352A	63DB	AF23	A0F9	2CF1	A925	4014
	E2	4		4		C		
T	585A	9E99	D2E	89D4	8FF3	55A9	05C2	C0AF
6	461D	D146	E592	705B	2FDF	A892	7BA	0BEF
2	BE63	BCB	7E8D	83B0	3D35	E350	A57C	3CBE
	EBB	3DC	352A	63DB	3775	A0F9	2CF1	A925
	B	E2	4				C	
	A	B	C	D	E	F	G	H
T	6453	585A	9E99	D2E	EB03	8FF3	55A9	05C2
6	A8D	461D	D146	E592	52A8	2FDF	A892	7BA
3	EC25	BE63	BCB	7E8D	78BD	3D35	E350	A57C
	2F0E	EBB	3DC	352A	CD4	3775	A0F9	2CF1
	2	B	E2	4	D			C
T	96EE	6453	585A	9E99	BD03	EB03	8FF3	55A9
6	C4FF	A8D	461D	D146	2DC	52A8	2FDF	A892
4	C680	EC25	BE63	BCB	ABF1	78BD	3D35	E350
	66C1	2F0E	EBB	3DC	23F6	CD4	3775	A0F9
		2	B	E2	F	D		
T	7A54	96EE	6453	585A	5E68	BD03	EB03	8FF3
6	E5F0	C4FF	A8D	461D	7CF3	2DC	52A8	2FDF
5	F8CE	C680	EC25	BE63	66C5	ABF1	78BD	3D35
	C511	66C1	2F0E	EBB	9BC6	23F6	CD4	3775
			2	B		F	D	
T	33FB	7A54	96EE	6453	EC67	5E68	BD03	EB03
6	DB49	E5F0	C4FF	A8D	FD80	7CF3	2DC	52A8
6	FDC	F8CE	C680	EC25	6F3C	66C5	ABF1	78BD
	BC15	C511	66C1	2F0E	F892	9BC6	23F6	CD4
	0			2			F	D
T	5CE0	33FB	7A54	96EE	5DFF	EC67	5E68	BD03
6	7D09	DB49	E5F0	C4FF	ED18	FD80	7CF3	2DC
7	2820	FDC	F8CE	C680	5A84	6F3C	66C5	ABF1
	CED	BC15	C511	66C1	4470	F892	9BC6	23F6
	A	0						F
T	FE94	5CE0	33FB	7A54	403D	5DFF	EC67	5E68
6	48C1	7D09	DB49	E5F0	70E7	ED18	FD80	7CF3
8	0200	2820	FDC	F8CE	E230	5A84	6F3C	66C5
	B89B	CED	BC15	C511	4AE2	4470	F892	9BC6
		A	0					
T	538A	FE94	5CE0	33FB	6C5F	403D	5DFF	EC67
6	D751	48C1	7D09	DB49	BD11	70E7	ED18	FD80
9	7A3C	0200	2820	FDC	29ED	E230	5A84	6F3C
	1812	B89B	CED	BC15	5D98	4AE2	4470	F892
			A	0				
T	F18B	538A	FE94	5CE0	674C	6C5F	403D	5DFF
7	A12	D751	48C1	7D09	2021	BD11	70E7	ED18
0	A8C8	7A3C	0200	2820	D74C	29ED	E230	5A84
	9F7F	1812	B89B	CED	4B3C	5D98	4AE2	4470
	3			A				
T	A253	F18B	538A	FE94	CD04	674C	6C5F	403D
7	44D7	A12	D751	48C1	EF73	2021	BD11	70E7
1	79BD	A8C8	7A3C	0200	9499	D74C	29ED	E230
	3C27	9F7F	1812	B89B	64CE	4B3C	5D98	4AE2
		3						
T	D674	A253	F18B	538A	F62C	CD04	674C	6C5F
7	1884	44D7	A12	D751	A4E6	EF73	2021	BD11
2	0227	79BD	A8C8	7A3C	0886	9499	D74C	29ED

	A	B	C	D	E	F	G	H
	76B6	3C27	9F7F3	1812	B853	64CE	4B3C	5D98
T7	722C	D674	A253	F18B	A869	F62C	CD04	674C
3	AB2E956	1884 0227	44D7 79BD	A12 A8C8	89CC 4784	A4E6 0886	EF73 9499	2021 D74C
	6918B	76B6	3C27	9F7F3	CC96	B853	64CE	4B3C
T7	3343	722C	D674	A253	8440	A869	F62C	CD04
4	666C4A22	AB2 E956	1884 0227	44D7 79BD	8D19 DCD	89CC 4784	A4E6 0886	EF73 9499
	A127	6918B	76B6	3C27	FA2FF	CC96	B853	64CE
T7	D620	3343	722C	D674	9DD8	8440	A869	F62C
5	0BB575E7	666C 4A22	AB2 E956	1884 0227	26E8 36FD	8D19 DCD	89CC 4784	A4E6 0886
	354E	A127	6918B	76B6	863B	FA2FF	CC96	B853
T7	2536	D620	3343	722C	1FA8	9DD8	8440	A869
6	0F7ACEF4	0BB5 75E7	666C 4A22	AB2 E956	A4FE 3190	26E8 36FD	8D19 DCD	89CC 4784
	CB4A	354E	A127	6918B	9744	863B	FA2FF	CC96
T7	A443	2536	D620	3343	35E1	1FA8	9DD8	8440
7	469B9040	0F7A CEF4	0BB5 75E7	666C 4A22	7267 8859	A4FE 3190	26E8 36FD	8D19 DCD
	BC0E	CB4A	354E	A127	93DF	9744	863B	FA2FF
T7	F469	A443	2536	D620	AC4F	35E1	1FA8	9DD8
8	6D63AEA	469B 9040	0F7A CEF4	0BB5 75E7	5582 3377	7267 8859	A4FE 3190	26E8 36FD
	5D9CA	BC0E	CB4A	354E	7F11	93DF	9744	863B
T7	2A4F	F469	A443	2536	38B2	AC4F	35E1	1FA8
9	E67FF1F8	6D63 AEA	469B 9040	0F7A CEF4	5046 A96B	5582 3377	7267 8859	A4FE 3190
	8289	5D9CA	BC0E	CB4A	321B	7F11	93DF	9744

7. Menghitung Variabel Kerja Dengan Inisial Nilai *Hash*

Tabel 8. Penjumlahan dengan *initial Hash value*

Variabel	*Initial Hash Value*		Variabel Kerja	Hasil
$H_0^{(0)}$	6A09E667F3BCC908	+	2A4FE67FF1F88289	9459CCE7E5B54B91
$H_1^{(0)}$	BB67AE8584CAA73B	+	F4696D63AEA5D9CA	AFD11BE933708105
$H_2^{(0)}$	3C6EF372FE94F82B	+	A443469B9040BC0E	E0B23A0E8ED5B439
$H_3^{(0)}$	A54FF53A5F1D36F1	+	25360F7ACEF4CB4A	CA8604B52E12023B
$H_4^{(0)}$	510E527FADE682D1	+	38B25046A96B321B	89C0A2C65751B4EC
$H_5^{(0}$	9B05688C2B3E6C1F	+	AC4F558233777F11	4754BE0E5EB5EB30
$H_6^{(0)}$	1F83D9ABFB41BD6B	+	35E17267885993DF	55654C13839B514A
$H_7^{(0)}$	5BE0CD19137E2179	+	1FA8A4FE31909744	7B897217450EB8BD

8. *Output*.

Output dari SHA 512 merupakan penggabungan dari $H_0^{(0)}$ sampai $H_7^{(0)}$ sebagai berikut :

9459CCE7E5B54B91 || AFD11BE933708105 || E0B23A0E8ED5B439 ||
CA8604B52E12023B || 89C0A2C65751B4EC || 4754BE0E5EB5EB30 ||
55654C13839B514A || 7B897217450EB8BD

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

9459CCE7E5B54B91AFD11BE933708105E0B23A0E8ED5B439CA8604B52E12023B89C0A2C65751B4EC4754BE0E5EB5EB3055654C13839B514A7B897217450EB8BD

4. IMPLEMENTASI

Kebutuhan sistem merupakan *spesifikasi* tentang perangkat yang akan digunakan yaitu *software* (perangkat lunak) dan *hardware* (perangkat keras) yang dibutuhkan oleh aplikasi. Berdasarkan kebutuhan sistem terdapat pengimplementasi suatu sistem yang merupakan proses menerjemahkan hasil analisa dan perancangan sistem kedalam sebuah bahasa pemrograman. Pengimplementasi suatu sistem akan berpengaruh pada *spesifikasi* komputer yang digunakan, agar program bisa berjalan dengan baik berdasarkan *spesifikasi* perangkat keras (*hardware*) dan perangkat lunak (*software*) harus sesuai. Sistem pengolahan suatu program terdiri dari prosedur dan pelaksanaan data, komputer sebagai sarana pengolahan *program* haruslah menyediakan fasilitas-fasilitas yang mendukung dalam pengolahan nantinya. Adapun kebutuhan sistem terbagi 2 yaitu perangkat keras dan perangkat lunak.

Perangkat keras adalah peralatan dari sistem komputer yang secara fisik terlihat dan dapat disentuh. Spesifikasi minimal perangkat keras *(Hadware*) yang digunakan untuk *implementasi* pembuatan dan menjalankannya, agar berjalan dengan baik adalah sebagai berikut:

1. *Prosessor* : minimal AMD A4-3330MX APU with Radeon(tm) HD Graphics 2.30 GHz
2. Memori : minimal 2 GB
3. *Harddisk* : minimal 100 GB
4. Monitor, Keyboard, dan Mouse.

Perangkat Lunak merupakan sebuah sistem yang digunakan untuk pengolah data atau aplikasi tertentu, sehingga perangkat lunak (*software)* yang digunakandapat dikategorikan sebagai berikut:

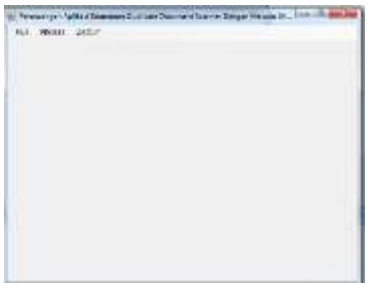
1. Sistem operasi merupakan suatu sistem untuk mengendalikan seluruh kegiatan yang dibutuhkan yang terdapat pada komputer, sehingga dalam sistem operasi yang nantinya digunakan yaitu sistem operasi minimal *windows* 7, *System Type* 32 Bit.
2. *Software* yang akan digunakan dalam menjalankan program berupa *software* bahasa pemrograman. Program yang digunakan untuk menerjemahkan *instruksi* yang ditulis dalam bahasa pemograman agar dapat dimengerti komputer. Berdasarkan pada sub bab sebelumnya untuk membangun atau merancang suatu program menggunakan aplikasi *Microsoft Basic Studio* 2008 dengan bahasa pemrograman *visual basic.*

A. Tampilan Pengujian

Tampilan program merupakan tampilan dari aplikasi dengan penerapan metode SHA-512 pada aplikasi pengamanan *duplicate document scanner* berbasis *desktop*. Aplikasi yang akan dijalankan telah dibangun dan dirancang

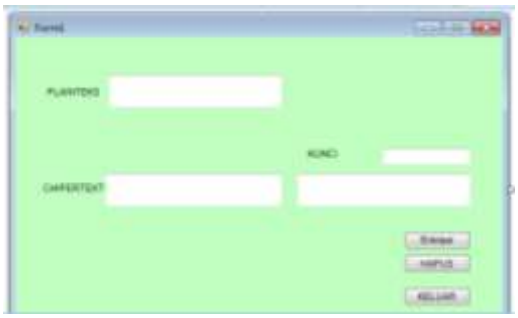
dengan menggunakan aplikasi *Microsoft Visual Studio* 2008 dengan bahasa pemrograman *visual basic 2008*.

Tampilan Menu Utama merupakan tampilan yang muncul pertama sekali pada saat program dijalankan. Tampilan ini berisi menu pendeteksian. Tampilan menu utama dapat dilihat pada Gambar 1.



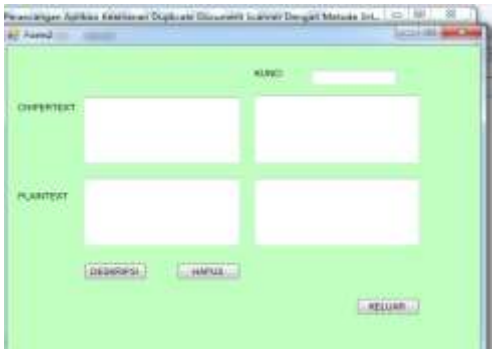
Gambar 1.Tampilan Menu Utama

Pada *form* ini akan ditampilkan proses enkripsi dengan menggunakan metode *SHA-512*. Adapun gambar proses tersebut dapat dilihat pada Gambar 2.



Gambar 2. Tampilan *Form* Enkripsi

Pada *form* ini akan ditampilkan proses deskripsi dengan menggunakan metode SHA 512. Adapun gambar proses tersebut dapat dilihat pada gambar 3. di bawah ini



Gambar 3. Tampilan Form Deskripsi

5. KESIMPULAN

Setelah dilakukan pengujian dan analisas sistem, Maka dapat diperoleh simpulan Dengan Penererapan Metode SHA-512 untuk pengamanan *duplicate document scanner* akan dapat menambah pengamanan pada *document* Penerapan metode pengamanan data pada *plainteks* dilakukan untuk menyembunyikan rahasia, yaitu dengan cara menerapkan algoritma simetris yang dilakukan melalui proses dekripsi. Dimana data pada *plainteks* awalnya diproses melalui kunci *enkripsi*, lalu data dialirkan menjadi kunci *dekripsi* sehingga akhirnya menjadi teks/data asli yang aman.

REFERENCES

[1] P. G. F. M. C. Q. F. R. Rigoni, ""Tampering detection of audio-visual content using encrypted watermarks,"," "Tampering detection of audio-visual content using encrypted watermarks,", p. 8, 2014.
[2] NIST, National Institute of Standart and TECHNOLOGY, American : Dede Djuhana, Ph.D, 1999.
[3] W. Dai, "Speed Benchmarks for Various Ciphers and Hash Functions.," Algoritma SHA-512, 2009.
[4] Yeni Agusti, "Perancangan," Perancangan Aplikasi Pembelajaran Fisika Tingkat, p. 6, 2013.
[5] Donny Yulianto, "Tahapan Perancangan," ANALISIS & PERANCANGAN SISTEM, p. 40, 2015.
[6] Mariyana, "Langkah-langkah dalam tahap perencanaan sistem," Langkah-langkah dalam tahap perencanaan sistem, p. 8, 2015.
[7] S. Sembiring, "Perancangan Sistem Informasi Pembelian Pada PT. Teknotama Lingkungan Internusa," Perancangan , p. 31, 2013.
[8] R. A. d. M. Shalahudin, " Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek," Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek, p. 10, 2014.
[9] P. P. Widodo, "UML," VOL.I NO.1 Februari 2015 Jurnal Teknik Komputer AMIK BSI11ISSN. 2442-2436 // Sistem Pakar Pendeteksian ....Sistem Pakar Pendeteksian Permasalahan Komputer Pada PT. Pasifik Satelit Nusantara Cikarang, 2011.
[10] Indraja, "Bagian alir (Flowchart)," Penggunaan Citra Penginderaan Jauh untuk Mendukung Mitigasi Dampak Perubahan Iklim di Sektor Pertanian, p. 14, 2015.
[11] Wicaksono, "Deteksi," Deteksi, p. 17, 2013.
[12] Nurasyiah, " Struktur File WAV dalam bentuk Hexa," Struktur File WAV dalam bentuk Hexa, p. 7, 2013.
[13] Nurasyiah, " Perancangan Aplikasi Kompresi File," Aplikasi Keamanan File Audio WAV (Waveform) Dengan, p. 7, 2013.
[14] D. Arius, "Pengantar Ilmu Kriptografi," Jurnal Teknik Informatika Kaputama (JTIK), Vol 1 No 1, Januari 2017ISSN :2548-97041 Perancangan Aplikasi Keamanan Pesan Menggunakan Algoritma Elgamal Dengan Memanfaatkan Algoritma One Timepadsebagai Pembangkit Kunci, p. 7, 2008.
[15] R. Sadikin, " Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java," Perbandingan, p. 5, 2012.
[16] R. Munir, " Kriptografi," Pemanfaatan Kriptografi Dalam Mewujudkan Keamanan Informasi Pada e-Voting Di Indonesia, p. 7, 2006.
[17] Novi dian Natasyah,anang eko wicaksono, "Penerapan Teknik Kriptografi Stream Cipher Untuk Pengaman Basis Data," Penerapan Teknik Kriptografi Stream Cipher Untuk Pengaman Basis Data, p. 22, 2011.
[18] D. Arius, "Pengantar Ilmu Kriptografi," Kriptografi File Citra Menggunakan, p. 23, 2018.