IMPLEMENTASI SHA512 PADA APLIKASI KRIPTOGRAFI FILE

Daurat Sinaga[1], Cahaya Jatmoko[2]
[1,2]Program Studi Teknik Infromatika, Fakultas Ilmu Komputer, Univresitas Dian Nuswantoro
e-mail: [1]daurat.sinaga@dsn.dinus.ac.id, [2]cahaya.jatmoko@dsn.dinus.ac.id

ABSTRAK

Permasalahan ini muncul bermula dari tindakan kriminal seperti halnya mencuri data atau files rahasia/penting (hacker) yang ada di desktop tanpa sepengetahuan pemilik data tersebut kemudian membuka data tersebut dan menyebarluaskan ke pihak yang tidak memiliki hak atas data atau files tersebut. Sehingga penelitian ini bertujuan untuk mengamankan data dari serangan hacker atau pihak yang tidak bertanggung jawab dan tidak mempunyai hak membuka dan menyebarluaskan dalam data atau files rahasia/penting tersebut. Dengan proses mengubah file kedalam format .enc, proses seperti ini dapat dinamakan sebagai mengenkripsi file agar data kita aman dari serangan hacker dengan menggunakan algoritma SHA-512 Hashing pada kriptografi dengan tools yang digunakan yaitu NetBeans IDE 8.0.2 serta menggunakan bahasa pemograman Java yang berbasis desktop ini. File yang diinputkan ke dalam aplikasi, lalu setelah itu user memasukan kunci yang dienkripsi dengan algoritma SHA-512 untuk mengamankan isi file yang akan diamankan. Data yang telah berhasil dienkripsi tidak mengurangi ukuran data dari file tersebut begitupun juga dengan setalah didekripsi.

***Kata Kunci**: Kriptografi, SHA512, Netbeans, File*

1. PENDAHULUAN

Kini perkembangan komputer dan teknologi internet sekarang seperti ini perkembangan digital dan teknologi informasi dengan kebutuhan informasi sangat dibutuhkan, sehingga sangat dibutuhkan tingkat keamanan dengan kerahasiaan data pada informasi tersebut. Keamanan yang dimiliki pada suatu data sangat diperlukan guna menjaga keaslian dan rahasia dalam informasi yang memiliki isi yang hanya dapat diketahui isi informasi rahasia tersebut oleh pihak yang berhak saja [1]. Pengiriman data maupun informasi yang tidak punya keamanan terhadap isi data ataupun informasi tersebut akan memiliki resiko terhadap serangan sadapan dalam isi informasi di dalamnya karna akan dapat sangat mudah untuk diketahui oleh pihak lain yang tidak berkepentingan pada informasi tersebut.

Sejalannya majunya perkembangan internet, di era digital sekarang merupakan era dimana berbagi informasi digital meningkat secara signifikan tentu saja akan memberikan dampak positif guna menyediakan pelayanan untuk mendapatkan informasi serta komunikasi, dari internet memberikan dampak negatif yang merupakan ancaman untuk pengguna layanan internet [2]. Oleh sebab itu keamanan yang terdapat dalam jaringan yang dimiliki komputer adalah bagian dari sebuah sistem terpenting guna pengamanan untuk menjaga kevaliditasian atas data tersebut serta integritas data yang bertujuan menjamin ketersediaan pelayanan untuk penggunanya, harus dilindungi dari segala serangan yang mungkin terjadi dan usaha untuk penyusupan atau pemindaian oleh pihak yang tidak berkewenangan dalam data tersebut. Keamanan menjadi hal yang sangat penting bagi semua pengguna namun tidak dapat dipungkiri bahwa jaminan keamanan masih lemah hingga saat ini. Pada 6 bulan awal tahun 2018 saja sudah ada 4,5 miliar data dicuri berdasarkan perusahaan Gemalto, jumlah ini meningkat sebanyak 113% dari tahun lalu, akan tetapi kasus pencurian data mengalami penurunan dari 1.162 kasus pada tahun lalu menjadi 945 kasus di tahun ini. Data yang diretas mencapai 6,9 juta data per harinya menurut laporan dari total 14,6 miliar sejak tahun 2013. Media sosial menjadi target terbesar para pihak ketiga yang tidak bertanggung jawab yaitu pencurian sebesar 56,11%, kemudian diikuti dengan data-data instansi pemerintah sebesar 26,62% yang sudah pasti sangat penting. Meskipun demikian hanya sebagian kecil dari data-data tersebut terlindungi dengan enkripsi yaitu hanya sekitar 4% saja [3]. Pada sistem komputer bisa disebut aman apabila suatu data hanya dapat diubah dan dibuka isi datanya oleh pihak yang telah diberi hak atas data tersebut.

Sehingga untuk meningkatkan tingkat keamanan yang terjadi pada lalulintas data dibutuhkan cabang ilmu khusus yang berisikan mengenai keamanan data atau biasa dinamakan sebagai ilmu kriptografi. Kriptografi yakni suatu cabang ilmu yang mengajarkan teknik - teknik yang mempunyai hubungan dengan aspek – aspek yang ada dalam keamanan informasi. Pada kripografi terdapat proses enkripsi dan dekripsi, yaitu proses penyandian dan pengembalian informasi [4]. Beberapa hal yang penting pada kerahasiaan serta keamanan data salah satunya dengan melakukan proses enkripsi. Enkripsi merupakan proses yang mengganti sebuah kode dari yang dapat dimengerti dirubah menjadi kode yang tidak dapat dimengerti oleh pembaca [5]. Pengenkripsian menghasilkan sebuah kode atau chiper.

Setiap orang pasti memiliki data penting yang harus mereka lindungi agar tidak hilang atau dicuri oleh pihak yang tidak bertanggungjawab. Terlebih pada suatu perusahaan yang memiliki lebih banyak data. Data yang dimiliki suatu perusahaan dapat menjadi aset yang memiliki sifat kerahasiaan yang digunakan untuk kelangsungan hidup perusahaan tersebut. Sebuah media penyimpanan data dengan sistem keamanan penuh

sangat diperlukan. Masalah yang timbul bukan hanya pada media penyimpanan data tetapi bagaimana media penyimpanan data tersebut dilengkapi dengan sistem keamanan yang baik pula. Oleh karena itu digunakanlah teknik enkripsi data. Dengan menggunakan teknik enkripsi sebagai sarana perlindungan informasi masalah dalam dokumen elektronik atau biasa disebut file ini dapat diatur ke setiap pengguna oleh enkripsi kunci milik pengguna itu sendiri, bahkan ketika berbagi komputer, orang lain tidak bisa dekripsi dengan benar dan tidak mendapatkan akses ke plaintext karena mereka tidak tahu kuncinya [6], sehingga menjamin keamanan file pribadi.

2. METODE PENELITIAN

2.1 Desain Antarmuka

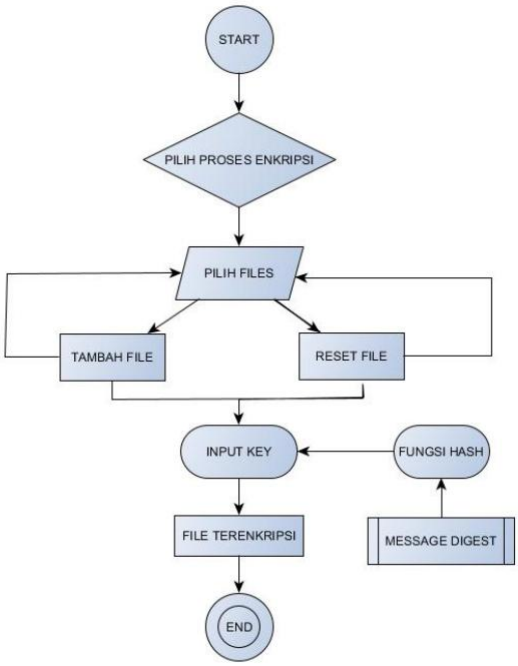
Perancangan interface adalah bagian terpenting dari aplikasi karena interface yang pertama dilihat ketika sedang menjalankan aplikasi. Perancangan interface aplikasi ini terdiri dari menu utama yang berisi dua pilihan aksi yaitu enkripsi dan pilihan dekripsi. Pembuatan interface menggunakan tools NetBeans IDE 8.0.2 dan bahasa pemograman Java. Kelas File Encryptor and Decryptor merupakan kelas yang digunakan untuk melakukan proses enkripsi dan dekripsi file. Proses pembuatan dalam program ini digunakan software NetBeans IDE 8.0.2 dengan menggunakan bahasa pemograman Java. Tampilan pada aplikasi File Encryptor and Decryptor diatur oleh folder Graphic User Interface (GUI). Tabel 1 adalah modul-modul yang terdapat dalam aplikasi dan beserta keterangan :

Tabel 1Module program file enkripsi dan dekripsi.

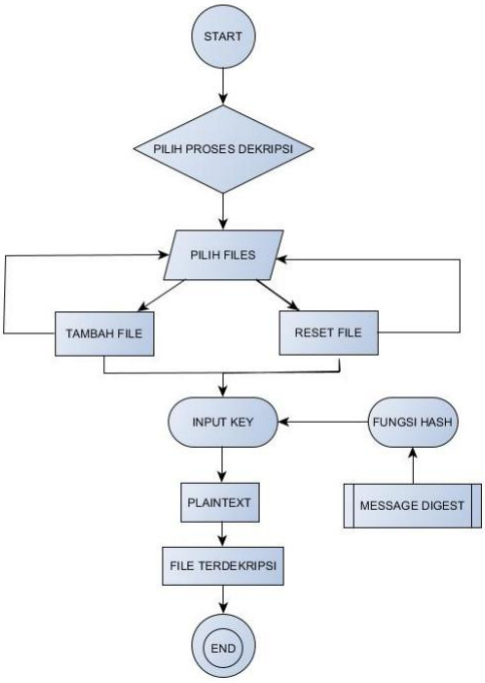
Module Name	Description
EncryptorAndDecryptor.Java	Activity untuk memanggil semua file pada directory(file manager yang ada di desktop) yang akan di enkripsi maupun yang akan didekripsi.
FileEncryptorAndDecryptor.Java	Memproses file yang dipilih untuk dienkripsi dengan menggunakan kunci.
EncryptorAndDecryptor Progress.Java	Activity yang menampilkan halaman utamayang digunakan untuk memilih proses yang akan dilakukan denganpilihan proses enkripsi dan dekripsi.
ExceptionDialog.Java	Activity yang mengatur jika terjadi error pada saat memproses file.
FileChooser.Java	Tampilan pada saat memilih file yang akan diproses enkripsi maupun dekripsi (directory files).
Main.Java	Tampilan interface yang menampilkan pada halaman utama program.
PasswordTakenForDecryption.Java	Tampilan untuk memasukan password yang akan di dekripsi menggunakan kunci.
PasswordTakenForEncryption.Java	Tampilan untuk memasukan password yang akan di enkripsi menggunakan kunci.
ThankYouDialog.Java	Tampilan akhir setelah mengakhiri aplikasi.

2.2 Prosedur Enkripsi dan Dekripsi

Pada Gambar 1, dilakukan perancangan sistem yang melakukan enkripsi file atau semua data multimedia yang menerapkan algoritma kriptografi SHA-512 Hashing untuk diterapkan kedalam aplikasi. Aplikasi program yang dibuat dengan proses enkripsi merupakan sebuah proses yang melakukan enkripsi file atau data yang ada pada desktop, dengan pertama yang dilakukan adalah pencarian file dan kemudian pemilihan file seperti yang ada pada dekstop seperti misalnya audio, dokumen, video, gambar, setup file dan lain sebagainya sebagai data plaintext/original.Setelah itu data dibaca oleh sistem jika sudah terbaca maka sistem akan melakukan ke langkah berikutnya yakni proses konversi kunci dan plaintextmenjadi bit array lalu kemudian proses diteruskan ke proses enkripsi. Proses enkripsi data yang berlangsung menggunakan algoritma SHA-512 dengan generate keamanan kunci. Kemudian data atau file rahasia telah aman dari serangan hacker yang akan mencuri ataupun membuka file rahasia yang disimpan



Gambar 1 Prosedur Proses Enkripsi File



Gambar 2. Prosedur Proses Enkripsi File

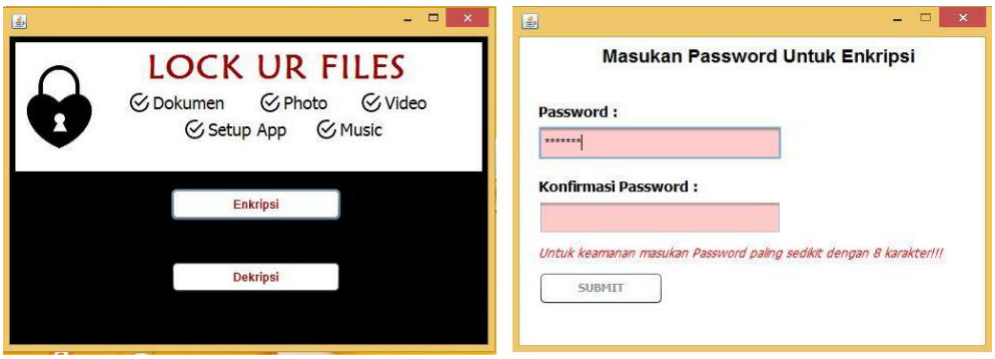
Gambar 2 digunakan sebagai alur dalam melakukan proses dekripsi file atau semua data multimedia. Proses dekripsi data ciphertext dari hasil proses enkripsi. Diawali dengan proses pengambilan data chiper file sebagai ciphertext, jika file sudah berhasil terbaca oleh sistem tanpa error maka proses dilanjutkan ke proses dekripsi. Setelah proses dekripsi berhasil file akan kembali ke file original.

4. HASIL DAN PEMBAHASAN

Pada tampilan menu utama terdapat bagian-bagian dari inti aplikasi yang telah dibuat. Fungsi-fungsi button dari bagian tersebut yakni meliputi :

a. Enkripsi : digunakan untuk memproses perintah enkripsi yang telah diberikan agar dieksekusi sehingga file tidak dapat terbaca.
b. Dekripsi : digunakan untuk mengembalikan file dalam bentuk semula sehingga file dapat terbaca kembali.

Pada interface di halaman pertama ini lah pengguna diminta memutuskan akan melakukan proses pengenkripsian file atau mendekripsi file.



Gambar 2 Desain Interface Aplikasi

Pengujian yang akan dilakukan terhadap aplikasi Lock Ur Files yang dibangun yaitu pengujian black box, white box, dan juga eksperimen sesuai Gambar 3. Jadi pengujian ini akan dilihat dari sisi interface atau bagian luar, kode program atau bagian dalam aplikasi, serta eksperimen untuk membuktikan berbagai ekstensi yang diteliti sesuai ekstensi yang ada di batasan masalah dapat bekerja dengan baik.

a. Pengujian Blackbox

Tabel 2 Pengujian Memasukkan Password

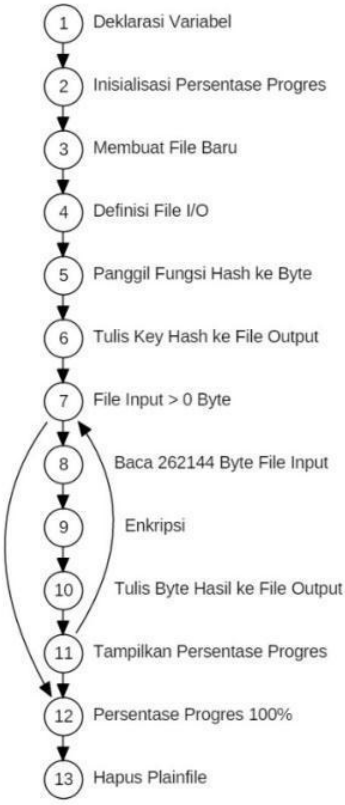
Kasus Input Data Normal			
Data Input	Output Harapan	Pengamatan	Kesimpulan
Kunci : hehehe30 Pengulangan Kunci: hehehe30	Kunci sesuai dan file dapat dieksekusi	Kunci memenuhi syarat, tombol enkripsi dapat diklik dan file dapat dieksekusi	Diterima
Kasus Input Data Tidak Normal			
Data Input	Output Harapan	Pengamatan	Kesimpulan
Kunci: 30he Pengulangan Kunci: 30he	Memberikan peringatan karakter kurang dari 8	Menampilkan keterangan bahwa minimal 8 karakter kunci dan tombol enkrip tidak bisa diklik	Diterima
Kunci: 30helo Pengulangan Kunci: helo30	Memberi pesan peringatan karena Kunci tidak sesuai	Menampilkan keterangan kunci tidak sesuai, tombol enkrip tidak bisa diklik, dan tidak dapat melanjutkan proses	Diterima

b. Pengujian Whitebox

Pengujian white box dilakukan pada sebuah aplikasi yang dibuat untuk mengecek source code secara detail dan dapat dijadikan acuan untuk memahami jalannya program atau logika yang digunakan pada program secara benar. Pertama pengujian yang dilakukan adalah pengujian basis path dimana logika akan digambarkan melalui sebuah diagram, selanjutnya menghitung Cyclomatic Complexity untuk mengetahui kompleksitas jalannya program dan untuk mengetahui jumlah jalur independen dimana jalur itu harus dipastikan dapat dieksekusi seluruhnya.

Berikut ini adalah proses pengujian proses enkripsi.

```
byte[] keyHash; ..... (1)
double percentageOfFileCopied=0;
byte[] buffer = new byte[262144];
progressArea.setText(progressArea.getText()+"  0%\n"); ..... (2)
destinationFile = new File(file.getAbsolutePath().concat(".enkrip")); .. (3)
if(destinationFile.exists()){
        destinationFile.delete();
        destinationFile = new File(file.getAbsolutePath().concat(".enkrip"));
}
BufferedInputStream fileReader = new
BufferedInputStream(newFileInputStream(file.getAbsolutePath())); ... (4)
FileOutputStream fileWriter = new FileOutputStream (destinationFile, true);
//encrypting content & writing
keyHash = getHash(key).getBytes(); ..... (5)
fileWriter.write(keyHash, 0, 128); ..... (6)
while(fileReader.available()>0){..... (7)
    int bytesCopied=fileReader.read(buffer); ..... (8)
    byte[] res = encryptBytes(buffer,key); ..... (9)
     fileWriter.write(res, 0, bytesCopied); ..... (10)
     long fileLength=file.length(); ..... (11)
        percentageOfFileCopied+=
   (((double)bytesCopied/fileLength)*100);
        showProgressOnprogressOfFilesTextField(progressArea,
   percentageOfFileCopied, bytesCopied, fileLength);
        showProgressOnProgressBarAndProgressPercent(progressBar,
    progressPercentage, bytesCopied, totalSizeOfAllFiles);
}
progressArea.setText(progressArea.getText().substring(0, progressArea.getText().length()-
5)+"100%\n"); ..... (12)
fileReader.close(); ..... (13)
fileWriter.close();
```



Gambar 3 Whitebox Proses Enkripsi

Perhitungan *Cyclomatic Complexity* adalah sebagai berikut:

V(G) = jumah edge – jumlah node + 2
= 14–13+2
= 3

V(G) = jumlah percabangan + 1
=2+1
= 3

Dari perhitungan di atas didapatkan bahwa V(G) dengan cara satu dan dua menunjukkan jumlah yang sama yaitu 3.

Sedangkan pengujian proses dekripsi seperti ditunjukkan code berikut.

```
String keyHash; ..... (1)
double percentageOfFileCopied=0;
byte[] kunci = new byte[128];
byte[] buffer = new byte[262144];
progressArea.setText(progressArea.getText()+"  0%\n"); ..... (2)
keyHash = getHash(key); ..... (3)
if (areHashesEqual(file, keyHash)){ ..... (4)
        destinationFile = new
          File(file.getAbsolutePath().toString().substring(0,
        file.getAbsolutePath().toString().length()-7)); ..... (5)
        BufferedInputStream fileReader=new BufferedInputStream
       (new FileInputStream(file.getAbsolutePath())); ..... (6)
        FileOutputStream fileWriter=new FileOutputStream
          (destinationFile);
        //decrypting content & writing
       fileReader.read(kunci); ..... (7)
       while(fileReader.available()>0){ ..... (8)
           int bytesCopied=fileReader.read(buffer); ..... (9)
             byte[] res = decryptBytes(buffer, key); ..... (10)
             fileWriter.write(res, 0, bytesCopied); ..... (11)
```

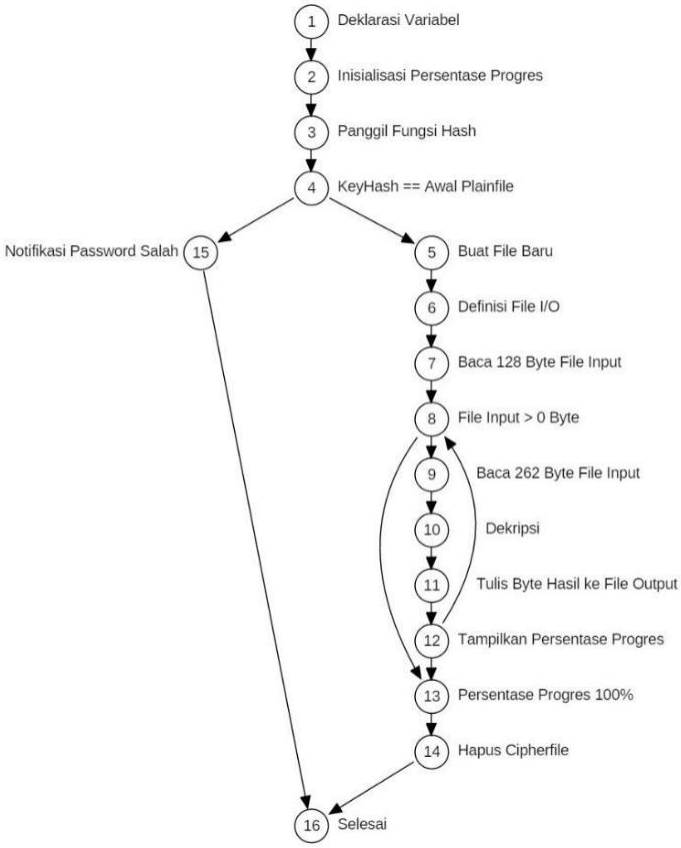
```
                long fileLength=file.length(); .....(12)
                percentageOfFileCopied+=
                (((double)bytesCopied/fileLength)*100);
                showProgressOnprogressOfFilesTextField(progressArea,
                percentageOfFileCopied, bytesCopied, fileLength);
                showProgressOnProgressBarAndProgressPercent
                (progressBar,progressPercentage, bytesCopied,
                totalSizeOfAllFiles);          }
            progressArea.setText(progressArea.getText().substring(0,
            progressArea.getText().length()-5)+"100%\n"); ..... (13)
            fileReader.close(); ..... (14)
            fileWriter.close();}
else if (!areHashesEqual(file, keyHash)) {
            progressArea.append("Invalid Password"); ..... (15)
}
 System.out.println("Proses Selesai"); ..... (16)
```

Perhitungan *Cyclomatic Complexity* adalah sebagai berikut:

V(G) = *jumah* edge – jumlah node + 2
= 18–16+2
= 4

V(G) = jumlah percabangan + 1
= 3 + 1
= 4

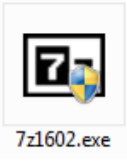
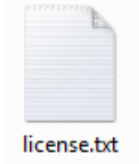
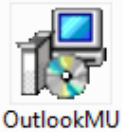
Dari perhitungan di atas didapatkan bahwa V(G) dengan cara satu dan dua menunjukkan jumlah yang sama yaitu 4 seperti ditunjukkan pada Gambar 5.

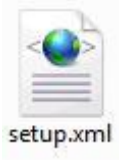


Gambar 5 White Box Dekripsi

Hasil Eksperimen terhadap file yang telah dienkripsi maupun dekripsi sesuai Tabel 3:

Tabel 3 Hasil eksperimen File

File asli		File enkripsi		File dekripsi		Kesimpulan
Plainfile	Ukuran plainfile	cipherfile	Ukuran cipherfile	plainfile	Ukuran cipherfile	
 7z1602.exe	1.05 MB (1,106,469 bytes)	 7z1602.exe	1.05 MB (1,110,016 bytes)	 7z1602.exe	1.05 MB (1,106,469 bytes)	Berhasil
 335.23-desktop-win8-win7-winvista-32bit-englis...	167 MB (175,616,000 bytes)	 335.23-desktop-win8-win7-winvista-32bit-english-whql	167 MB (175,612,496 bytes)	 335.23-desktop-win8-win7-winvista-32bit-englis...	167 MB (175,616,000 bytes)	Berhasil
 AdbeRdr11010_en_US.exe	72.3 MB (75,858,112 bytes)	 AdbeRdr11010_en_US.exe	72.3 MB (75,862,016 bytes)	 AdbeRdr11010_en_US.exe	72.3 MB (75,858,112 bytes)	Berhasil
 DWA-123_D1_FW_4.03.zip	60.7 MB (63,733,760 bytes)	 DWA-123_D1_FW_4.03.zip	60.7 MB (63,733,630 bytes)	 DWA-123_D1_FW_4.03.zip	60.7 MB (63,733,760 bytes)	Berhasil
 README.HTM	4.00 KB (4,096 bytes)	 README.HTM	4.00 KB (4,092 bytes)	 README.HTM	4.00 KB (4,096 bytes)	Berhasil
 license.txt	4.00 KB (4,096 bytes)	 license.txt	4.00 KB (4,087 bytes)	 license.txt	4.00 KB (4,096 bytes)	Berhasil
 OutlookMUI.msi	2.01 MB (2,117,632 bytes)	 OutlookMUI.xml	2.01 MB (2,115,584 bytes)	 OutlookMUI.msi	2.01 MB (2,117,632 bytes)	Berhasil

	8.00 KB (8,192 bytes)		8.00 KB (8,201 bytes)		8.00 KB (8,192 bytes)	Berhasil

Pada Eksperimen aplikasi Lock Ur Files sesuai Gambar 3, file yang terenkripsi ukurannya meningkat namun tidak signifikan/sedikit. Meskipun ukurannya meningkat namun isi file yang didalamnya tidak berubah. Ukuran file pada file asli dan file hasil dekripsi tidak berubah sama sekali, dengan demikian dapat dinyatakan bahwa proses dekripsi berhasil. Seluruh proses enkripsi menghasilkan file yang tidak dikenal dan tidak dapat dibuka seperti halnya file yang telah dikunci.

5. KESIMPULAN

Pada penelitian kali ini peneliti dapat menyimpulkan beberapa hal penting yaitu sebagai berikut :

- Berhasil membangun software yang diberi nama Lock Ur File berbasis desktop untuk mengenkripsi file yang mengimplementasikan metode SHA-512 untuk verifikasi kunci untuk mengamankan sebuah file pribadi/penting.
- Software berhasil mengenkripsi dan mendekripsi file dokumen teks, gambar, audio, maupun video dengan ekstensi .jpg, .bmp, .jpeg, .png, .mp4, .mkv, .wmv, .mp3, .txt, .pdf, .doc, .docx, .pptx, .xlsx dan .exe.
- Ukuran file hasil dari proses enkripsi bertambah besar dari file aslinya akan tetapi ukuran kembali semula seperti file aslinya setelah melalui proses dekripsi.
- Metode akan lebih sulit dipecahkan oleh kriptanalis dengan alasan langkah untuk mengenkripsi cukup banyak karena menggunakan lebih dari satu metode selain itu kriptanalis tidak akan menyadari bahwa bit awal dari file yang terenkripsi merupakan hash kunci hasil fungsi SHA-512, terlebih lagi hash kunci tidak ditulis sepanjang 512 bit atau 64 byte melainkan 128 byte.

DAFTAR PUSTAKA

[1] Mukti Qamal, "Kriptografi File Citra Menggunakan Algoritma TEA (Tiny Encryption Algorithm)," *TECHSI*, vol. 6, no. 2, 2014.

[2] Leonardo Refialy, Eko Sediyono, and Adi Setiawan, "Pengamanan Sertifikat Tanah Digital menggunakan Digital Signature SHA-512 dan RSA," *JuTISI*, vol. 3, no. 1, Desember 2015.

[3] Agustin Setyo Wardani. (2018, Oktober) Liputan 6.com. [Online]. https://www.liputan6.com/tekno/read/3665291/45-miliar-data-dicuri-selama-6-bulan-pertama-2018

[4] Tri Ferga Prasetyo and Aris Hikmawan, "Analisis Perbandingan Dan Implementasi Sistem Keamanan Data Menggunakan Metode Enkripsi RC4 SHA Dan MD5," *Infotech Journal*, vol. 2, no. 1, 2016.

[5] Fitri Nuraeni, Yoga Handoko Agustin, and Irman Maulana Muharam, "Implementasi Tanda Tangan Digital Menggunakan RSA dan SHA-512 Pada Proses Legalisasi Ijazah," in *Konferensi Nasional Sistem Informasi (KNSI)*, Pangkalpinang, 2018, pp. 864-869.

[6] Alam Rahmatulloh, Heni Sulastri, and Rizal Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *JNTETI*, vol. 7, no. 2.