Jurnal Maklumatika
Vol. 9, No. 1, Juli 2022 ISSN : 2407-5043

PERANCANGAN APLIKASI DATA SECURITY DALAM MELINDUNGI INFORMASI DIGITAL MENGGUNAKAN TEKNIK ALGORITMA RIJNDAEL BERBASIS DESKTOP

Adhitya Ahmad Pradypta

Program Studi Teknik Informatika, Universitas Islam As-Syafi'iyah, Bekasi, Indonesia
Jalan Jatiwaringin Raya Nomor 12, Jaticempaka, Pondok Gede, Bekasi, Jawa Barat
Email: adhitya.fst@uia.ac.id

Abstrak

Seperti yang kita ketahui saat ini ilmu pengetahuan dan teknologi berkembang dengan sangat cepat, penggunaan berbagai macam informasi dan juga media yang menggunakan teknologi multimedia saat ini digunakan setiap harinya mulai dari personal / individu bahkan sampai organisasi maupun istitusi perusahaan. Salah satu Contohnya adalah aktivitas berfoto, merekam video dan lain-lain. Namun dibalik keunggulan perkembangan teknologi yang cepat saat ini juga terdapat beberapa celah keamanan khususnya pada data yang dimiliki oleh masing-masing individu atau lembaga. Sehingga perlu adanya suatu penelitian untuk mengamankan data - data penting yang akan digunakan baik oleh individu, instansi pemerintah/swasta, sekolah ataupun universitas. Salah satu hal pengamanan data yang bisa dilakukan adalah implementasikan sebuah metode enkripsi pada aplikasi pengamanan data. Hasil penelitian ini berupa aplikasi yang dibuat untuk mengamankan data–data penting seperti file yang berekstensi doc, exe, mp3, swf, mp4, avi dan lain – lain. Simpulan penelitian ini semua jenis format file dapat dilakukan menggunakan enkripsi dengan baik, file yang telah terenkripsi akan ditandai dengan file yang berekstensi *.encrypt. Aplikasi data security berbasis desktop ini dirancang dapat bekerja dengan baik dengan menggunakan teknik algoritma rijndael.

***Kata kunci**: Aplikasi enkripsi, dekripsi, algoritma, rijndael, desktop.*

1. PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi merupakan suatu perangkat yang banyak digunakan oleh seseorang untuk mengolah data, mendapatkan informasi, dan lain sebagainya. Keberadaan teknologi informasi tentu saja akan berpengaruh terhadap lingkungannya di masyarakat serta memberikan dampak yang negatif dan positif khususnya yang terkait dengan privasi data digital. (Zulfah, 2018)

Privasi merupakan hal yang sangat penting baik bagi setiap individu, organisasi atau lembaga dalam berinteraksi dengan individu lain atau dengan lembaga lain. Apabila salah dalam menyampaikan sesuatu informasi digital yang kemungkinannya memiliki nilai *confidential, classified* bahkan sangat rahasia, tentu saja tidak dapat di pungkiri akan sangat menyebabkan kerugian secara material ataupun non material. Apalagi jika terdapat informasi digital yang disampaikan merupakan rahasia yang berisikan peta kekuatan atau strategi yang dirancang dalam menghadapi persaingan dengan kompetitor bahkan terlebih lagi yang berkaitan dengan organisasi. Informasi pribadi yang tidak ingin di berikan, dan diketahui oleh umum, apabila sudah terlanjur, dan tersebar maka hal tersebut akan membahayakan posisi kredibilitas dari individu atau lembaga tersebut. (Yuwinanto, 2011)

Berbicara tentang keamanan adalah suatu keadaan terbebas dari bahaya. Keamanan juga dapat diimplementasikan di berbagai hal, termasuk juga data dan informasi digital. Data merupakan sebuah aset penting dalam kehidupan baik secara individu bahkan lembaga. Semua orang tentunya memikirkan data dan informasi yang dimiliki terjaga keamanannya dan tidak mudah terpublikasi ke khalayak umum. (Ade Chandra Saputra, 2019)

Data maupun informasi digital merupakan aset yang penting, apabila aset tersebut tidak dijaga, maka dapat menimbulkan suatu kerugian baik materil maupun immateril. Keamanan data dan informasi digital dapat berupa file seperti teks, gambar, suara, atau video. Tentu saja dengan

perkembangan teknologi, data, dan informasi dapat disajikan secara digital. Namun bentuk penyimpanannya masih sangat rentan dari segi aspek keamanannya. Data bisa saja dengan mudah dihilangkan, dimanipulasi, bahkan disalahgunakan oleh orang yang tidak bertanggung jawab. Dari permasalahan tersebut tentu saja melanggar hak privasi seseorang. Dengan demikian pengamanan data digital ini menjadi hal yang sangat penting sekali dan sangat mendesak untuk di implementasikan. (Saputra & Agus Sehatman Saragih, 2020)

Adapun sebuah solusi dalam masalah keamanan data dan informasi digital yaitu dengan menerapkan sebuah teknik kriptografi. Kriptografi merupakan teknik untuk mencegah kebocoran data dan informasi digital yang bersifat rahasia. (Amrulloh & EIH.Ujianto, 2019) Pengamanan data dan informasi digital dapat dilakukan dengan algoritma enkripsi dan dekripsi. Enkripsi data adalah mengubah data awal menjadi suatu informasi baru yang telah disamarkan dengan menggunakan kunci. Dekripsi data adalah mengembalikan seperti informasi awal. Salah satu kriptografi *advanced encryption standard* (AES) dengan teknik Algoritma yang dapat diimplementasikan yaitu dengan teknik algoritma kriptografi aes rinjdael. Jenis kriptografi aes rijndael ini cukup handal hingga saat ini. Pada tahun 2006, *National Security Agency* (NSA) pernah menyatakan bahwa kriptografi AES cukup aman digunakan dalam mengamankan data dan informasi pemerintah Amerika Serikat saat itu yang bukan tergolong sangat rahasia. (Zendrato, 2019)

1.2. Rumusan Masalah

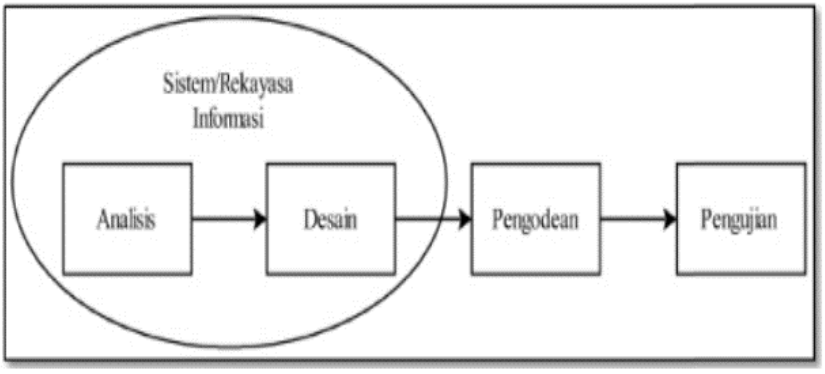
Berdasarkan latar belakang masalah yang telah dipaparkan diatas maka ada beberapa hal yang menjadi pokok permasalahan, antara lain sebagai berikut:

1. Bagaimana merancang aplikasi untuk mengamankan *file* dengan cara enkripsi dekripsi menggunakan teknik algoritma rijndael?
2. Apakah hasil dekripsi dapat mengembalikan file hasil enkripsi ke bentuk awal?

2. METODOLOGI

Metode penelitian yang digunakan untuk melakukan penelitian adalah:

1. Penelitian kepustakaan
   Menurut Mirzaqon (2017) penelitian kepustakaan merupakan kegiatan penelitian yang dilakukan melalui pengumpulan informasi dan data di perpustakaan seperti buku referensi, hasil penelitian sebelumnya yang sejenis, artikel, catatan, serta berbagai jurnal yang berkaitan dengan masalah yang ingin dipecahkan. (Mirzaqon. T, 2017)
2. Metode Pengembangan Sistem
   Pada tahapan metode pengembangan sistem ini dilakukan model rekayasa sistem menggunakan *System Development Life Cycle* (SDLC) merupakan gambaran dari suatu bentuk usaha dalam tahapan proses pengembangan sistem. (Abdullah, 2017) Alat bantu yang digunakan dalam Metode *System Development Life Cycle* banyak sekali namun dalam penelitian ini yang digunakan adalah model waterfall. Model SDLC air terjun (waterfall) sering disebut juga model sekuensial linier (sequential linear) atau alur hidup klasif (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (support). (Arisantoso et al., 2022) Berikut adalah gambar model air terjun:



Gambar 1. Model *Waterfall*

3. HASIL DAN PEMBAHASAN

3.1. Enkripsi

Enkripsi adalah suatu proses untuk merubah sebuah pesan, data atau informasi (biasa disebut *plaintext*), sehingga informasi tersebut tidak bisa dibaca oleh orang yang tidak bertanggung jawab (*cipherteks*). Jadi *plainteks* adalah informasi yang dapat di mengerti dan *cipher* teks adalah informasi yang tidak dapat dimengerti atau tidak dapat dibaca. Sebuah sistem pegkodean menggunakan sebuah table atau kamus yang telah didefinisikan untuk mengganti kata dari informasi yang dikirim. Sebuah *cipher* menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) *bit* dari sebuah pesan menjadi *cryptogram* yang tidak di mengerti (*unnitelligible*). Karena teknik cipher merupakan suatu sistem yang telah siap untuk diautomasi, maka teknik ini digunakan dalam sistem keamanan komputer dan jaringan.

Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemenantara kedua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan cipherteks, maka fungsi enkripsi E memetakan P ke C.

$E(P) = C$

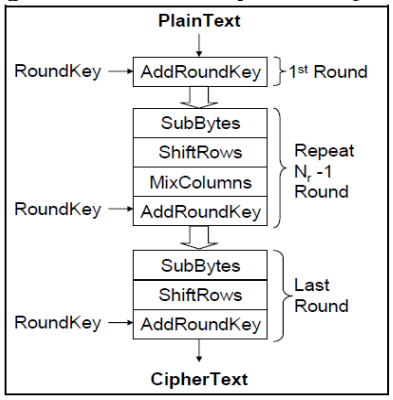
Dan fungsi dekripsi D memetakan C ke P,

$D(C) = P$

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka kesamaan berikut harus benar,

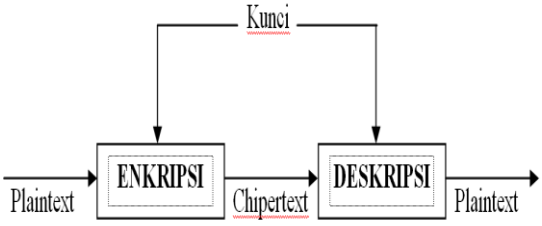
$D(E(P)) = P$

Keamanan algoritma sering diukur dari banyaknya kerja (*work*) yang dibutuhkan untuk memecahkan *ciphertext* menjadi plainteksnya tanpa mengetahui kunci yang digunakan. Kerja ini dapat diekuivalenkan dengan waktu, memori, uang,dan lain-lain. Semakin banyak kerja yang diperlukan, yang berarti juga semakin lama waktu yang dibutuhkan, maka semakin kuat algoritma kriptografi tersebut, yang berarti semakin aman digunakan untuk menyandikan pesan.



Gambar 2. Diagram Proses Enkripsi *Rijndael*.

Enkripsi *atau Cipher* berlangsung dalam rentetan empat fungsi pembangun (primitif) yaitu: *SubByte(), ShiftRows(), MixColumns(),* dan *AddRoundKey().* Rentetan tersebut dijalankan sebanyak Nr-1 sebagai *loop* utama. Setelah loop utama tersebut berakhir (sembilan *round*), *SubByte(), ShiftRows(),* dan *AddRoundKey().* Dieksekusi secara berturut-turut sebagai *final round.*



Gambar 3. Proses Enkripsi Dekripsi Secara Umum

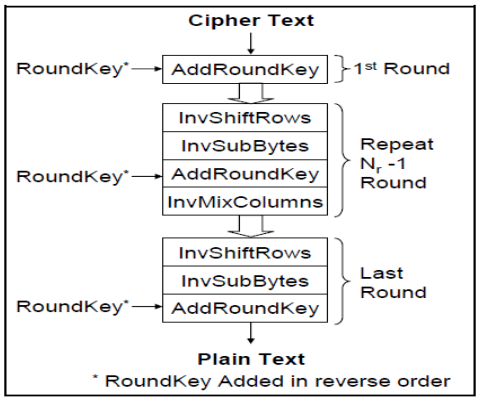
Enkripsi juga digunakan untuk *verifikasi*. Dalam hal ini terdapat tiga katagori enkripsi, yaitu

1. Kunci enkripsi rahasia. Dalam hal ini, terdapat sebuah kunci yang digunakan untuk mengenkripsi dan juga sekaligus mendekripsi informasi.
2. Kunci enkripsi publik. Dalam hal ini, dua kunci digunakan. Satu untuk proses enkripsi dan yang lain untuk proses dekripsi.

Fungsi *one-way*, atau fungsi satu arah adalah susatu fungsi dimana informasi senkripsi untuk menciptakan signature dari informasi asli yang bisa digunakan untuk keperluan autentikasi. Enkripsi digunakan berdasarkkan suatu algoritma yang akan mengacak suatu informasi menjdi bentuk yang tidak bisa dibaca atau tidak bisa dilihat.

3.2. Dekripsi

Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikan informasi teracak menjadi asli. Algoritma yang digunakan harus terdiri dari susunan prosedur yang direncanakan secara hati-hati yang harus secara efektif menghasilkan sebuah bentuk ter-enkripsi yang tidak bisa dikenalikan oleh seseorang, bahkan sekalipun mereka memiliki algoritma yang sama. Transformasi-transformasi yang merupakan kebalikan dari setiap *cipher* diterapkan dalam program dekripsi (*inverse cipher*). Fungsi *AddRoundKey()* untuk enkripsi digunakan kembali untuk dekripsi. Adapun yang harus dibuat lagi adalah *InvSubBytes()*, *InvShiftRows()*, dan *InvMixColumns()*. Beberapa bagian cukup dikopi dari fungsi kebalikannya yang telah digunakan saat enkripsi. *AddRoundKey()* dieksekusi sebagai *initial round*, diikuti sembilan round rentetan *InvShiftRows()*, *InvSubBytes()*, *InvMixColumns()*, dan *AddRoundKey()*. Round ke-10 yang mengikutinya tidak menyertakan *InvMixColumns* serupa dengan final round enkripsi.



Gambar 4. Diagram Proses Dekripsi *Rijndael*.

3.3. Algoritma Rijndael

Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan *cipher block*. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. *Rijndael* mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun *Rijndael* mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 *bit*. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

Tabel 1. Jumlah Proses Berdasarkan *Bit* Blok Dan Kunci

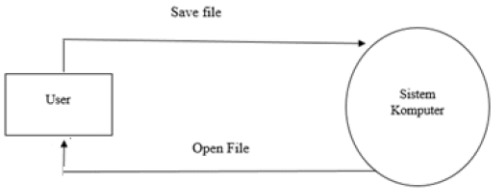
Panjang Kunci (NK) Dalam Words	Ukuran Blok Data (Nb) Dalam Words	Jumlah proses (Nr)
4	4	10
6	4	12
8	4	14

Blok-blok data masukan dan kunci dioperasikan dalam bentuk *array*. Setiap anggota *array* sebelum menghasilkan keluaran *ciphertext* dinamakan dengan *state*. Setiap *state* akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, *AddRoundKey, SubBytes, ShiftRows,* dan *MixColumns*. Kecuali tahap *MixColumns*, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap *MixColumns* tidak akan dilakukan pada tahap terakhir. Proses enkripsi adalah kebalikkan dari dekripsi. Karena terjadi beberapa tahap dalam proses enkripsi, maka diperlukan *subkey-subkey* yang akan dipakai pada tiap tahap. Pengembangan jumlah kunci yang akan dipakai diperlukan karena kebutuhan *subkey-subkey* yang akan dipakai dapat mencapai ribuan *bit*, sedangkan kunci yang disediakan secara *default* hanya 128-256 *bit*. Jumlah total kunci yang diperlukan sebagai *subkey* adalah sebanyak *Nb(Nr+1),* dimana *Nb* adalah besarnya blok data dalam satuan *word*. Sedangkan *Nr* adalah jumlah tahapan yang harus dilalui dalam satuan *word*. Sebagai contoh, bila mana digunakan 128 *bit* (4 *word*) blok data dan 128 *bit* (4 *word*) kunci maka akan dilakukan 10 kali proses (lihat Tabel 1). Dengan demikian dari rumus didapatkan 4(10+1)=44 *word* =1408 *bit* kunci. Untuk melakukan pengembangan jumlah kunci yang akan dipakai dari kunci utama maka dilakukan *key schedule*.

3.4. Rancangan Data Flow Diagram Sistem Usulan

Berikut ini dijelaskan rancangan diagram alur data sistem usulan aplikasi enkripsi dekripsi.

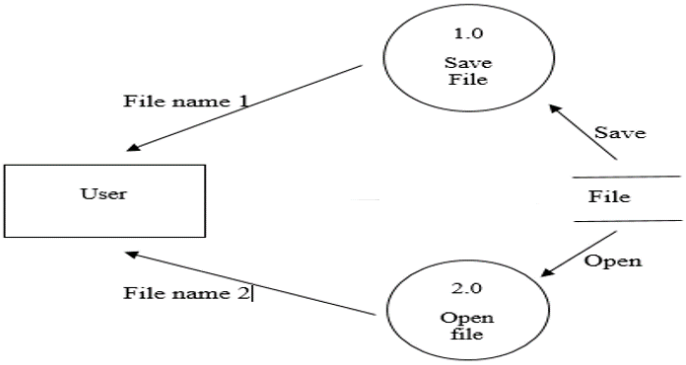
1. Diagram Konteks



Gambar 5. Diagram Konteks Sistem Usulan

Jika dilihat pada gambar 5 diatas terdapat satu sistem terminator yaitu *user* yang akan melakukan *save file* untuk diproses di sistem komputer dan juga proses *open file* yang akan dikembalikan lagi ke *user*.

2. Diagram Nol



Gambar 6. Diagram Nol Sistem Usulan

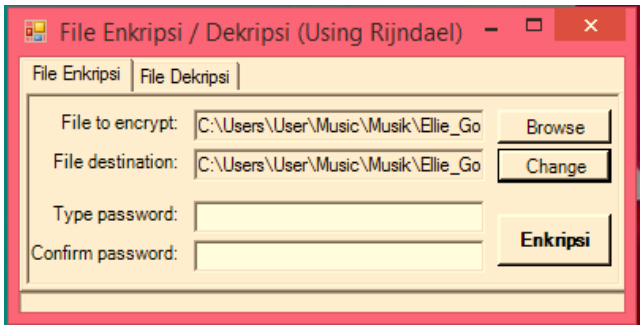
Pada gamabr 6 terdapat terminator yang bernama *user*, simbol *file* dan dua proses yang mana user memasukan *file* name untuk melakukan proses pertama yaitu *save file* dan setelah itu *user* bisa melakukan proses kedua yaitu *open file.*

3.5. Proses Enkripsi

Gambar 7. *Form* Enkripsi

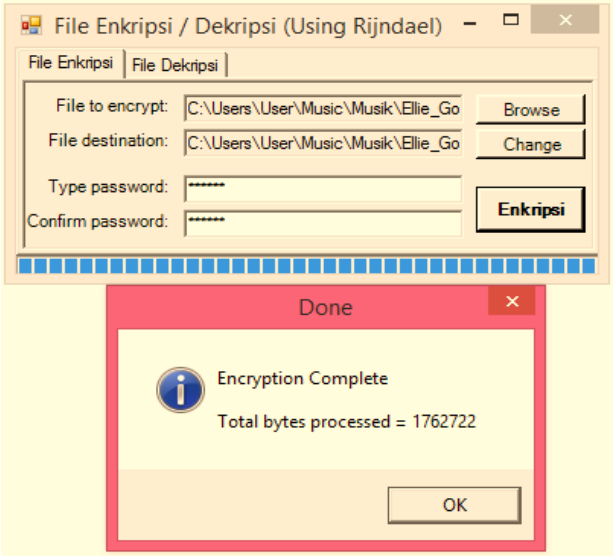
Tampilan File enkripsi ini akan menampilkan fungsi-fungsi seperti gambar 7 diatas terdapat *button browse*, *change*, enkripsi. Gunanya untuk menjalankan proses enkripsi hanya dengan klik *button* file enkripsi setelah itu klik *button browse* lalu akan muncul halaman sebagai berikut:

Pilih salah satu jenis *file* yang akan di enkripsi berbentuk *file*.



Gambar 8. *Input File* Enkripsi.

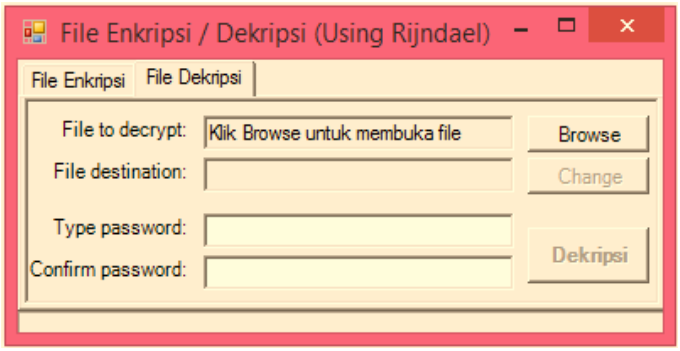
Setelah memilih file yang akan dienkripsi klik *browse* untuk memilih jenis *file* yang akan dienkripsi untuk ditempatkan di bagian folder yang dipilih. Contohnya seperti gambar 8 diatas.



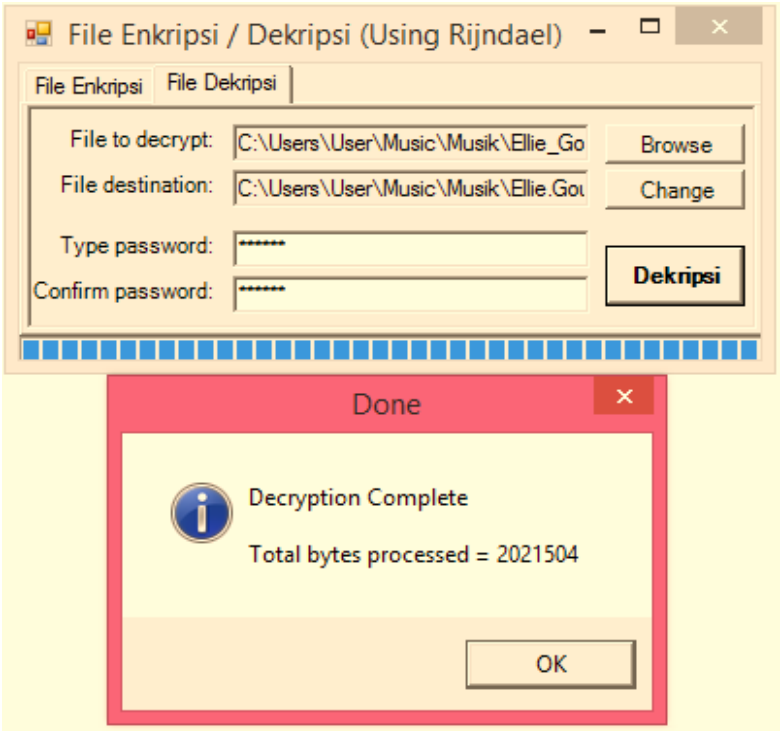
Gambar 9. Proses Enkripsi Telah Berhasil

Proses enkripsi file yang telah berhasil di jalankan, terdapat pesan dengan jumlah *bytes* yang telah di proses.

3.6. Proses Dekripsi

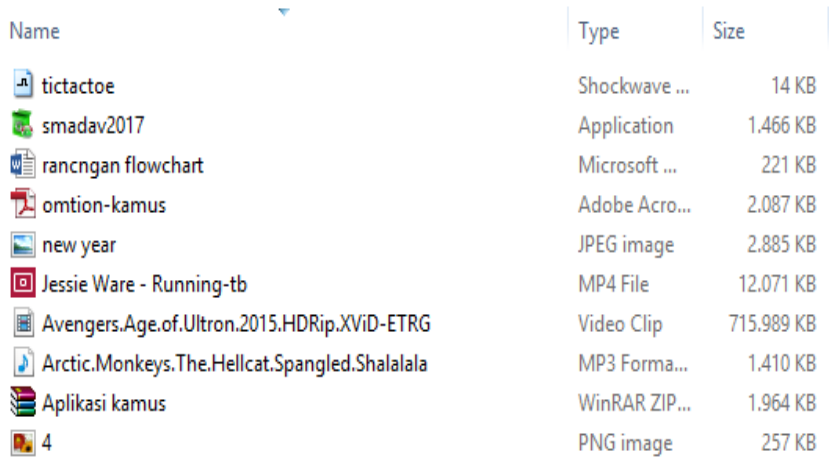


Gambar 10. *Form* Dekripsi.

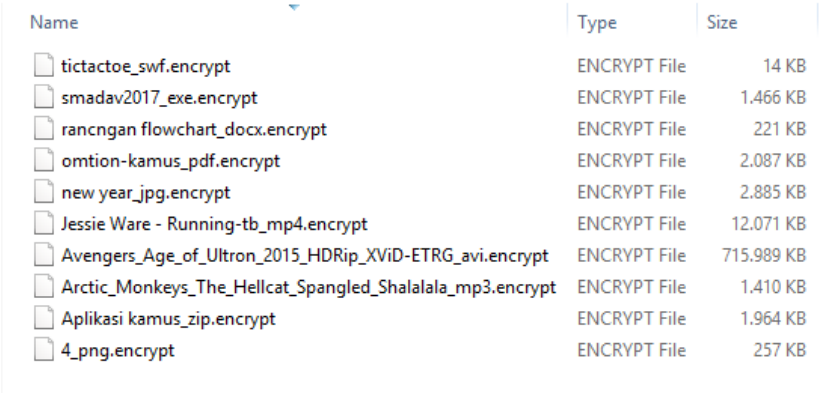


Gambar 11. Proses dekripsi Telah Berhasil

Hampir sama dengan prosesnya seperti enkripsi, untuk menjalankan proses dekripsi pengguna hanya perlu memilih jenis *file* dekripsi, selanjutnya akan tampil partisi pada *harddisk* untuk memilih file enkripsi yang akan di kembalikan *file* seperti semula atau dengan teknik dekripsi. Selanjutnya memilih klik *change* untuk menempatkan *file* yang akan di dekripsi. Masukan kunci yang pertama kali digunakan sebahgai proses enkripsi setelah selesai masukan juga kunci ke dalam *textbox* pada kolom *confirm password*, setelah itu tampil dan klik dekripsi untuk menjalankan proses dekripsi. Sama hal nya seperti proses enkripsi pada aplikasi ini untuk menjalankan proses dekripsi hanya bisa digunakan satu *file* per proses.



Gambar 12. Contoh File Sebelum Di Proses Enkripsi



Gambar 13. Contoh *File* Setelah Di Proses Enkripsi

3.7. Pengujian Aplikasi Dengan Metode *Black Box System*

Tabel 2. Pengujian Aplikasi

Nama File	Kapasitas File	Hasil Kinerja Enkripsi	Kapasitas Enkripsi	Hasil Kinerja Dekripsi
Tictactoe.swf	14 kb	Berhasil terenkripsi	14 kb	Berhasil terdekripsi
Smadav2017.exe	1.466 kb	Berhasil terenkripsi	1.466 kb	Berhasil terdekripsi
Rancangan flowchart.doc	221 kb	Berhasil terenkripsi	221 kb	Berhasil terdekripsi
Omtion-kamus.pdf	2.087 kb	Berhasil terenkripsi	2.087 kb	Berhasil terdekripsi
New year.jpg	2.885 kb	Berhasil terenkripsi	2.885 kb	Berhasil terdekripsi
Jessie ware_running.mp4	12.071 kb	Berhasil terenkripsi	12.071 kb	Berhasil terdekripsi
Avenger age of ultron 2015.avi	715.989 kb	Berhasil terenkripsi	715.989 kb	Berhasil terdekripsi
Arctic monkey-the hellcat.mp3	1.410 kb	Berhasil terenkripsi	1.410 kb	Berhasil terdekripsi
Aplikasi kamus.zip	1.964 kb	Berhasil terenkripsi	1.964 kb	Berhasil terdekripsi
4.png	257 kb	Berhasil terenkripsi	257 kb	Berhasil terdekripsi

4. KESIMPULAN

Berikut kesimpulan yang dapat diambil dari aplikasi yang telah di rancang sebagai berikut:

1. Aplikasi dirancang menggunakan bahasa pemrograman visual studio dengan *File* yang telah terenkripsi berekstensi (*.*encrypt*), dapat dibuktikan melalui hasil pengujian dengan melihat gambar 13.
2. Hasil nilai akurasi program sebesar 100% dan semuanya berhasil dienkripsi dan dikembalikan kebentuk awal dengan dekripsi.

DAFTAR PUSTAKA

Abdullah, D. (2017). Merancang Aplikasi Perpustakaan Menggunakan SDLC. *CV. Sefa Bumi Persada.*

Ade Chandra Saputra, A. S. S. (2019). Rancang Bangun Website Badan Pengawas Pemilihan Umum (Bawaslu) Kalimantan Tengah. *Jurnal Teknologi Informasi*, *13*(1), 9–10.

Amrulloh, A., & EIH.Ujianto. (2019). Kriptografi Simetris Menggunakan Algoritma Vigenere Cipher. *Jurnal CoreIT*, *5*(2), 71.

Arisantoso, Harjanti, T. W., & Yulianti, S. D. (2022). Modul Pembelajaran Rekayasa Perangkat Lunak. In *CV. Eureka Media Aksara* (pp. 41–42).

Mirzaqon. T, A. dan B. P. (2017). Studi Kepustakaan Mengenai Landasan Teori dan Praktik Konseling Expressive Writing. *Jurnal BK Unesa*, *8*(1).

Saputra, A. C., & Agus Sehatman Saragih. (2020). Implementasi Algoritma Rijndael Dalam Enkripsi Dan Dekripsi Gambar Digital Berbasis Web. *Jurnal Teknologi Informasi*, *14*(1), 52–53.

Yuwinanto, H. P. (2011). Privasi Online dan Keamanan Data. *Jurnal Palimpsest*, *2*(2).

Zendrato, W. A. P. (2019). Implementasi Algoritma Rijndael Untuk Pengamanan Pada File Video. *Jurnal Pelita Informatika*, *7*(4), 564.

Zulfah, S. (2018). Pengaruh Perkembangan Teknologi Informasi Lingkungan (Studi Kasus Kelurahan Siti Rejo I Medan). *Jurnal Buletin Utama Teknik*, *13*(2).