IMPLEMENTASI METODE RSA DAN AES UNTUK MENGAMANKAN FILE WINRAR DAN ZIP

IMPLEMENTATION OF RSA AND AES METHODS TO SECURE WINRAR AND ZIP FILES

Rizki Nanda Nasution[1], Budi Triandi[2]
[1]Jurusan Teknik Informatika, Fakultas Teknik dan Imu Komputer
[2]Dosen Jurusan Teknik Informatika Universitas Potensi Utama
[1,2]Universitas Potensi Utama, Jl.K.L. Yos Sudarso Km. 6,5 No 3A Tanjung Mulia Medan
E-mail: [1]rizky.nanda1997@gmail.com,[2]buditriandi@gmail.com

ABSTRAK

Perkembangan teknologi informasi yang pesat menjadikan informasi menjadi permintaan utama setiap orang atau setiap organisasi, karena informasi sangat penting untuk membantu individu atau organisasi berkembang dalam persaingan global. Demi menjamin keamanan file tersebut, agar tidak jatuh ke tangan orang lain, biasanya seseorang memberikan password kepada Rar dan Zip yang mereka miliki. Rar dan Zip adalah file terkompresi, yang dapat menggabungkan beberapa file menjadi ukuran yang lebih kecil. Agar file lebih aman, telah ditambahkan dua metode enkripsi yaitu metode RSA (Rivest-Shamir-Adleman) dan metode AES (Advanced Encryption Standard). RSA adalah salah satu teknik kriptografi dimana kunci untuk melakukan enkripsi berbeda dengan kunci untuk melakukan dekripsi. sedangkan AES adalah block chiphertext simetrik yang dapat mengenkripsi (encipher) dan dekripsi (decipher) sehingga dapat lebih mudah untuk menentukan pasangan kunci public dan private yang di buat secara random dan tidak bisa di gunakan untuk file yang lain.

Kata Kunci : *Rar dan Zip, Enkripsi Rsa dan Aes.*

ABSTRACT

The rapid development of information technology makes information the main demand of every person or every organization, because information is very important to help individuals or organizations develop in global competition. In order to ensure the security of these files, so they don't fall into the hands of other people, usually someone gives the passwords to Rar and Zip they have. Rar and Zip are compressed files, which can combine multiple files into a smaller size. In order to make files more secure, two encryption methods have been added, namely the RSA (Rivest-Shamir-Adleman) method and the AES (Advanced Encryption Standard) method. RSA is a cryptographic technique where the key for encryption is different from the key for decryption. while AES is a symmetric ciphertext block that can encrypt (encipher) and decrypt (decipher) so that it can be easier to determine public and private key pairs that are randomly generated and cannot be used for other files..

Keywords: *Rar and Zip, Rsa And Aes Encryption.*

1. PENDAHULUAN

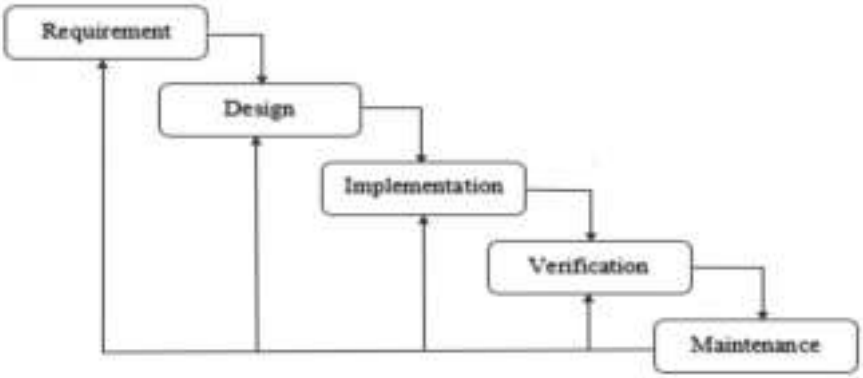
Keamanan file adalah suatu cara untuk melindungi file dari ancaman, baik dalam bentuk kesengajaan maupun tidak disengaja. Kemajuan teknologi informasi yang sangat pesat membuat permasalahan keamanan sering bermunculan terutama dalam keamanan file. Dengan adanya keterangan di atas maka dibutuhkan sebuah cara untuk mengatasi keamanan file tersebut, salah satu cara pengamanan file dalam dunia teknologi komputer dan jaringan adalah dengan metode kriptografi. Kriptografi bertujuan untuk menyamarkan data atau informasi yang dikomunikasikan. Kriptografi terbagi atas dua yaitu, kriptografi klasik dan kriptografi *modern*.

Namun pada kasus ini keamanan file yang akan dibahas adalah keamanan file pada aplikasi berbasis dekstop. Oleh karena itu, penulis tertarik untuk melakukan penelitian tentang sistem keamanan berbasis teknologi enkripsi RSA dan AES untuk melindungi keamanan file WinRAR dan zip, karena metode RSA dan AES belum pernah digunakan untuk melindungi keamanan file winrar dan zip sebelumnya, biasanya pengamanan file winrar dan zip hanya menggunakan password saja sehingga dapat dengan mudah untuk di crack (di bobol),. Aplikasi yang penulis buat nantinya memiliki fitur enkripsi dan dekripsi file. Demi meningkatkan keamanan file penulis menggunakan kombinasi algoritma untuk menjaga keamanan file agar lebih terjamin dari serangan-serangan yang dapat membahayakan isi file yang tersimpan.

Berdasarkan penelitian yang dilakukan Yovie Prsetio, Budi Triandi, Hardianto. (2018). Untuk melindungi dan menjaga rahasia data untuk menghindari orang yang tidak berhak mendapatkan informasi ini, yaitu menggunakan metode kriptografi. Metode kriptografi memiliki teknik dan metode mereka sendiri. Salah satu metode enkripsi yang dapat digunakan adalah dengan menggunakan skema hybrid dari teka-teki kriptografi dan RSA, dimana enigma adalah algoritma simetris yang digunakan untuk melindungi pesan asli (teks biasa), dan RSA adalah algoritma asimetris yang digunakan untuk melindungi kunci [1].

Berdasarkan Penelitian yang di lakukan oleh Tulloh, A. R., Permanasari, Y., & Harahap, E. (2016), membahas tentang enkripsi file dokumen dengan melakukan enkripsi perblok (128 bit) seara pararel untuk memudahkan enkripsi maupun dekripsi di gunakan algoritma RSA dan software aplikasi Matlab[2].

2. METODE PENELITIAN



Gambar 1. Prosedur Perancangan Dengan Metode Waterfall.

Metode *waterfall* memiliki tahap-tahap metode pengembangan dalam perancangan aplikasi ini, antara lain sebagai berikut:

1. *Requirements System,* pada tahap ini, layanan sistem, batasan, dan tujuan ditentukan oleh hasil konsultasi dengan pengguna, kemudian pengguna didefinisikan secara detail untuk memahami sistem. Perancangan Sistem dan Perancangan Perangkat Lunak Pada tahap ini sistem mengalokasikan kebutuhan sistem dengan membentuk arsitektur sistem secara keseluruhan, termasuk perangkat keras dan perangkat lunak. Desain perangkat lunak melibatkan mengidentifikasi dan menjelaskan abstraksi sistem perangkat lunak dasar dan hubungannya.
2. *Implementation,* pada tahap ini, seluruh desain sistem yang telah disiapkan sebelumnya akan diubah menjadi kode program dan modul, serta akan diintegrasikan ke dalam sistem yang lengkap sesuai kontrak..
3. *Integration & System Testing,* ditahap ini proses ujicoba terhadap aplikasi yang telah dirancang, pengujian ini perlu dilaksanakan untuk mengetahui apakah aplikasi sudah berjalan sesuai dengan baik yang telah ditetapkan*Operation & Maintenance,* tahapan ini adalah tahap terahir dari seluruh rangkaian kegiatan perancangan aplikasi, dimana pada tahap ini terjadinya proses perbaikan sistem seperti terjadi kesalahan-kesalahan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dan layanan sistem sebagai kebutuhan baru.

3. HASIL DAN PEMBAHASAN

3.1. Metode

Penerapan metode *RSA* dan *AES* untuk mengamankan file Winrar dan Zip adalah sebagai berikut:

1. *Riverst Shamir Adleman* (RSA)
   RSA adalah algoritma yang digunakan untuk enkripsi kunci publik. Algoritme ini adalah algoritme pertama yang dikenal paling sesuai untuk tanda tangan dan enkripsi, dan penemuan besar pertama dalam kriptografi kunci publik. RSA masih banyak digunakan dalam protokol e-commerce dan dianggap sangat aman karena diberi kunci panjang dan implementasinya mutakhir [3].
   Contoh pembentukan kunci Algoritma RSA :
   Diketahui bahwa p dan q merupakan dua bilangan prima yang besar yaitu bilangan acak dan bilangan rahasia, p ≠ q, dan p dan q berukuran sama. -Hitung n = p x q, lalu hitung ϕ (n) = (p-1) x (q-1) .Bilangan bulat n disebut modulus (RSA). -Menentukan e bilangan prima acak dengan ketentuan sebagai berikut: 1 <e <1 (n), PBT (e, i (n)) = 1, disebut e relatif terhadap i (n) bilangan prima, disebut integer n (RSA) Komponen terenkripsi, jadi Dd (Ee (m)) = Ee (Dd (c)) ≡md mod n [4].
2. *Advanced Encryption Standart(AES).*
   AES adalah sistem pengkodean non-pseudo-blok yang menggunakan penggantian, teknik permutasi dan banyak siklus untuk setiap blok yang akan dienkripsi [5].
   Contoh:
   Plainteks: KRIPtografi AES Kunci: Rizki Nanda Nst1
   Maka plainteks ini dapat dibuat menjadi state berikut:
   Plainteks :

Tabel 1. Plainteks

K	t	A	A
R	o	F	E
I	g	I	S
P	r	Space	Null

Kunci :

Tabel 2. Kunci

R	I	N	N
I	Space	D	S
Z	N	A	T
K	A	Space	1

konversi menjadi bilangan heksa desimal dengan KODE *ASCII* seperti berikut:

Tabel 3. *Konversi Plainteks* Heksa ke *ASCII*

4B	74	61	41
72	6F	66	45
69	67	69	53
70	72	20	00

Tabel 4. *Konversi* Kunci Heksa ke *ASCII*

52	69	6E	4E
69	20	65	73
7A	4E	61	74
6B	61	20	31

Proses Ekspansi Kunci seperti berikut:

Tabel 5. Prose Ekspansi Kunci

52	69	6E	4E
69	20	65	73
7A	4E	61	74
6B	61	20	31

Sebelum memasuki babak awal atau babak 0, kita harus terlebih dahulu mencari kunci bulat untuk setiap babak. Di babak 0, subKey sama dengan kunci yang dimasukkan pertama kali. Pada putaran 0, teks biasa akan melalui proses AddRounKey Pada tahap ini, perhitungan XOR dilakukan antara teks biasa dan kunci. Contoh matriks teks [1,1] matriks kunci XOR [1,1] dan matriks teks [2,1] matriks kunci XOR [2,1], dan seterusnya. Dibawah ini akan disimulasikan proses XOR antara baris ke-1 kolom ke-1 pada masing -masing matriks.

Baris ke-1 kolom ke-1 dari matriks plainteks : 4B
Baris ke-1 kolom ke-1 dari matriks kunci : 52

Untuk melakukan XOR terlebih dahulu ubah bentuknya dari hexa ke bentuk biner sehingga seperti dibawah ini.

Baris ke-1 kolom ke-1 dari matriks plainteks 4B (biner 8 bit) : 1001011
Baris ke-1 kolom ke-1 dari matriks kunci 52 (biner 8 bit) : 01010010
hasil nya 1001011 + 01010010 = 0011001 (19)
Baris ke-2 kolom ke-2 dari matriks plainteks 74 (biner 8 bit) : 1110100
Baris ke-2 kolom ke-2 dari matriks kunci 69 (biner 8 bit) : 1101001
Hasilnya 1110100 + 1101001 = 0011101 (1D)
Baris ke-3 kolom ke-3 dari matriks plainteks 61 (biner 8 bit) : 1100001
Baris ke-3 kolom ke-3 dari matriks kunci 6E (biner 8 bit) : 1101110
Hasilnya 1100001 + 1101110 = 0001111 (F)
Baris ke-4 kolom ke-4 dari matriks plainteks 41 (biner 8 bit) : 1000001
Baris ke-4 kolom ke-4 dari matriks kunci 4E (biner 8 bit) : 1001110
Hasilnya 1000001 + 1001110 = 0001111 (F)
Baris ke-5 kolom ke-5 dari matriks plainteks 72 (biner 8 bit) : 1110010
Baris ke-5 kolom ke-5 dari matriks kunci 69 (biner 8 bit) : 1101001
Hasilnya 1110010 + 1101001 = 0011011 (1B)
Baris ke-6 kolom ke-6 dari matriks plainteks 6F (biner 8 bit) : 1101111
Baris ke-6 kolom ke-6 dari matriks kunci 20 (biner 8 bit) : 0100000
Hasilnya 1101111 + 1101111 = 1001111 (4F)
Baris ke-7 kolom ke-7 dari matriks plainteks 66 (biner 8 bit) : 1100110
Baris ke-7 kolom ke-7 dari matriks kunci 65 (biner 8 bit) : 1100101
Hasilnya 1100110 + 1100101 = 1001111 (3)
Baris ke-8 kolom ke-8 dari matriks plainteks 45 (biner 8 bit) : 1000101
Baris ke-8 kolom ke-8 dari matriks kunci 73 (biner 8 bit) : 1110011
Hasilnya 1000101 + 1110011 = 0110110 (36)
Baris ke-9 kolom ke-9 dari matriks plainteks 69 (biner 8 bit) : 1101001
Baris ke-9 kolom ke-9 dari matriks kunci 7A (biner 8 bit) : 1111010
Hasilnya 1101001 + 1111010 = 0110110 (13)
Baris ke-10 kolom ke-10 dari matriks plainteks 67 (biner 8 bit) : 1100111
Baris ke-10 kolom ke-10 dari matriks kunci 4E (biner 8 bit) : 1001110
Hasilnya 1100111 + 1001110 = 0101001 (29)
Baris ke-11 kolom ke-11 dari matriks plainteks 69 (biner 8 bit) : 1101001
Baris ke-11 kolom ke-11 dari matriks kunci 61 (biner 8 bit) : 1100001
Hasilnya 1101001 + 1100001= 0101001 (8)
Baris ke-12 kolom ke-12 dari matriks plainteks 53 (biner 8 bit) : 1010011
Baris ke-12 kolom ke-12 dari matriks kunci 74 (biner 8 bit) : 1110100

Hasilnya 1010011+ 1110100= 0100111 (27)
Baris ke-13 kolom ke-13 dari matriks plainteks 70 (biner 8 bit) : 1110000
Baris ke-13 kolom ke-13 dari matriks kunci 6B (biner 8 bit) : 1101011
Hasilnya 1010011+ 1101011= 0011011 (1B)
Baris ke-14 kolom ke-14 dari matriks plainteks 72 (biner 8 bit) : 1110010
Baris ke-14 kolom ke-14 dari matriks kunci 61 (biner 8 bit) : 1100001
Hasilnya 1110010+ 1100001= 0010011 (13)
Baris ke-15 kolom ke-15 dari matriks plainteks 20 (biner 8 bit) : 0100000
Baris ke-15 kolom ke-15 dari matriks kunci 20 (biner 8 bit) : 0100000
Hasilnya 0100000 + 0100000= 0000000 (0)
Baris ke-16 kolom ke-16 dari matriks plainteks 00 (biner 8 bit) : 0000000
Baris ke-16 kolom ke-16 dari matriks kunci 31 (biner 8 bit) : 0110001
Hasilnya 0000000 + 0110001 = 0110001 (31)
maka akan mendapatkan hasil seperti ini :

Tabel 6. Bentuk Biner Dari Matriks

19	1D	F	F
1B	4F	3	36
13	29	8	27
1B	13	0	31

Sampai tahap ini ronde 0 telah selesai, selanjutnya akan dilanJutkan ronde ke-1.Pembentukan subKey berhubungan dengan *subkey* yang terbentuk pada ronde sebelumnya. Semua dijelaskan pada tabel dibawah ini.

Tabel 7. Proses *SubKey*

W i-4			W1-1	W i										
52	69	6E	4E											
69	20	65	73											
7A	4E	61	74											
6B	61	20	31											
0				1				2			3			

Untuk memperoleh kolom ke-i dari matriks 4x4 diperlukan kolom ke i-4 dan i-1 dari *round key* yang telah ada. Khusus kolom pertama pada tiap *round key* cara perhitungannya berbeda dengan cara perhitungan pada kolom 2, 3 dan 4. Pertama, ambil kolom Wi-1 kemudian *cell* paling atas dipindah ke cell paling bawah atau istilah *RotWord.*

Tabel 8. Proses *Round Key*

4E	Pergeseran	73
73		74
74		6B
6B		4E

Setelah itu hasilnya di *subsitusi* dengan *S-Box* Hasil substitusi dengan *S-Box* menjadi seperti berikut :

Tabel 9. Proses *Subtitusi* dengan *S-Box*

4E	S-Box	2F
73		8F
74		92
6B		7F

Setelah disubstitusi, maka hasil tersebut akan di-XOR dengan kolom Wi-4 dan kolom pertama dari tabel Rcon Tentunya tata cara XOR juga cell terhadap cell yang urutan baris dan kolomnya sama. Hasil XOR terlihat pada gambar dibawah.

Tabel 10. Proses *XOR Cell* per *Cell*

52	Xor	2F	Xor	01	Hasil	7C
69		8F		00		E6
7A		92		00		E8
6B		7F		00		14

Setelah diperoleh kolom pertama dari *subkey* ke-1 maka tabel *subKey* akan terlihat seperti berikut.

Tabel 11. Hasil Proses *SubKey* ke-1

W i-4			W1-i	W i-1	W i									
52	69	6E	4E	7C										
69	20	65	73	E6										
7A	4E	61	74	E8										
6B	61	20	31	14										
0				1				2			3			

Sekarang kita akan mencari kolom ke-2 dari *subkey* ke-1, dimana juga diperlukan kolom i-1 dan i-4 dari tabel *subKey*. Cara mencari kolom ke-2 lebih sederhana dibandingkan dengan cara mencari kolom pertama, cukup men-XOR-kan kolom i-1 dan i-4 saja. Hasilnya akan seperti ini.

Tabel 12. Proses Mencari Kolom ke-2 *SubKey* ke-1

7C	Xor	69	Hasil Xor	15
E6		20		C6
E8		4E		A6
14		61		75

Sekarang mencari kolom ke-2 dari *subkey* ke-1, Cara mencari kolom ke-2 cukup men-XOR-kan kolom i-1 dan i-4 saja. untuk mendapatkan kolom ke-3 dan ke-4 caranya sama kolom yang ke-2. Sehingga hasil keseluruhannya akan seperti ini.

Tabel 13. Hasil Proses Mencari Kolom ke-2 *SubKey* ke-1

52	69	6E	4E	7C	15									
69	20	65	73	E6	C6									
7A	4E	61	74	E8	A6									
6B	61	20	31	14	75									
0				1				2			3			

Setelah selesai mendapatkan *subKey* untuk ronde ke-1 kita akan lanjut ke proses subBytes. Pada bagian ini kita melakukan proses substitusi *S-Box* terhadap hasil *addRoundKey* pada ronde ke-0. Dimana hasilnya adalah sebagai berikut.

Tabel 14. Proses *subBytes*

19	1D	F	F	XOR	D4	A4	8C	8C
1B	4F	3	36		AF	84	04	05
13	29	8	27		7D	A5	CD	CC
1B	13	0	31		AF	7D	63	C7

Selanjutnya adalah proses *shiftRows* hasil *shiftRows* terhadap hasil *subBytes* yang diperoleh diatas adalah seperti berikut.

D4	A4	8C	8C			D4	A4	8C	8C
AF	84	04	05	Rotasi	`	05	84	04	AF Rotasi Satu Byte
7D	A5	CD	CC			CD	CC	7D	A5 Rotasi Dua Bytes
AF	7D	63	C7			C7	AF	7D	63 Rotasi Tiga Bytes

Selanjutnya adalah proses *mixColumns*, pada proses ini, terjadi perkalian matriks 4x4 antara hasil *shiftRows* yang diperoleh diatas dengan matriks.

02	03	01	01	matriks 4x4	D4	A4	8C	8C	hasil perkalian nya	w11	w12	w13	w14
01	02	03	01		05	84	04	AF		w21	w22	w23	w24
01	01	02	03		CD	CC	7D	A5		w31	w32	w33	w34
03	01	01	02		C7	AF	7D	63		w41	w42	w43	w44

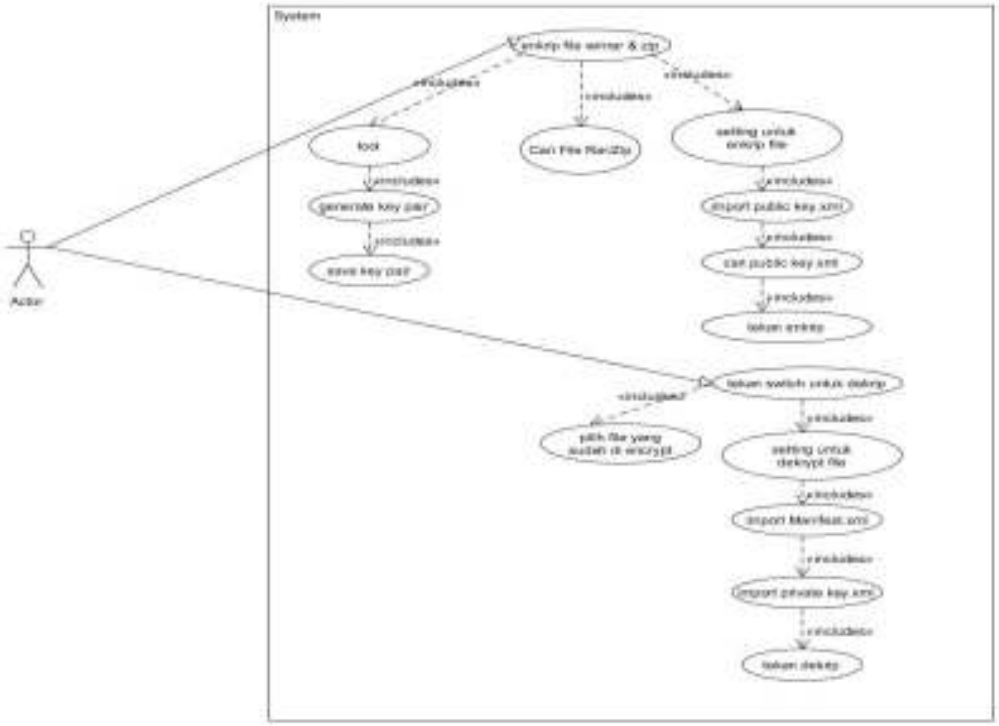
Sama seperti perkalian matriks biasa, untuk mendapatkan W11, rumusnya adalah:

W11 = (02 * D4) ⊕(03 * 05) ⊕(01 * CD) ⊕(01 * C7),
W12 = (02 * A4) ⊕(03 * 84) ⊕(01 * CC) ⊕(01 * AF),
W13 = (02 * 8C) ⊕(03 * 04) ⊕(01 * 7D) ⊕(01 * 7D),
W14 = (02 * 8C) ⊕(03 * AF) ⊕(01 * A5) ⊕(01 * 63), dan seterusnya.

untuk ronde ke-2 sampai ke-9 ulangi saja proses diatas, dan untuk ronde terakhir atau ronde ke-10, lewati proses *mixColumns*. Setelah semua proses selesai dilakukan maka akan di dapat hasil dekripsi : 4B 72 69 70 74 6F 67 72 61 66 69 20 41 45 33 00

3.2.*Use Case Diagram*

Diagram use case adalah diagram yang menjelaskan secara singkat siapa yang menggunakan sistem dan apa yang dapat dilakukan olehnya.

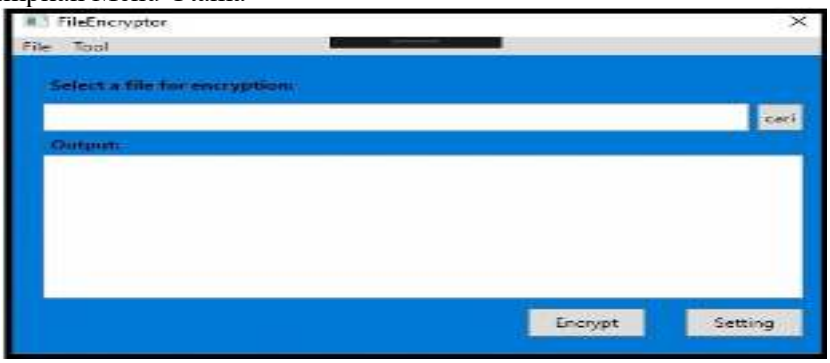


Gambar 2. *Use Case Diagram*

3.3. Tampilan Aplikasi

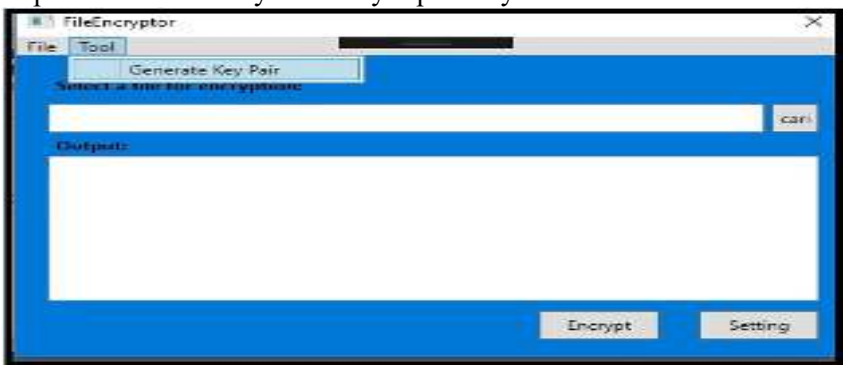
Berikut ini adalah tampilan hasil dari program Implementasi Metode *Riverst Shamir Adleman* (*RSA*) dan *Advanced Encryption System(AES)* untuk mengamankan File Winrar dan Zip.

1. Tampilan Menu Utama



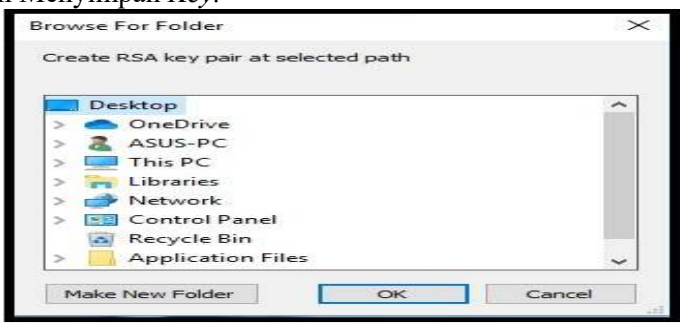
Gambar 3. Tampilan Menu Utama.

2. Tampilan Membuat Key Dan Meyimpan Key.



Gambar 4. Tampilan Membuat Key.

3. Tampilan Menyimpan *Key.*

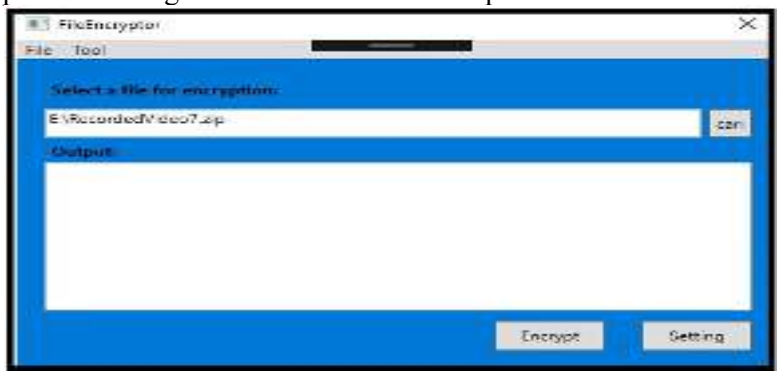


Gambar 5. Tampilan Menyimpan *Key.*

4. Tampilan Cari Untuk Men *Encrypt File.*

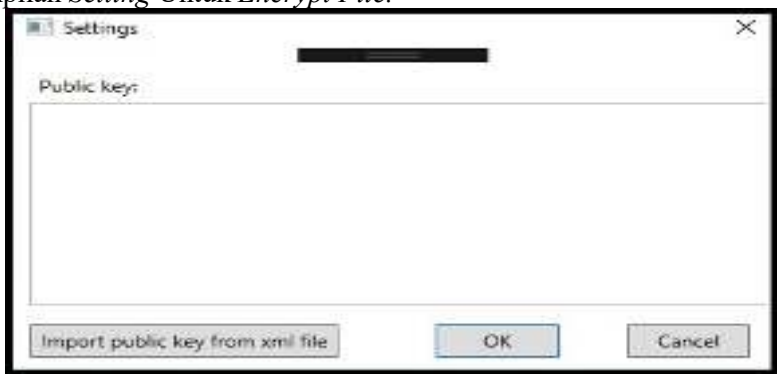
Gambar 6. Tampilan Cari *File.*

5. Tampilan file Yang Sudah Dimasukkan Ke Aplikasi.



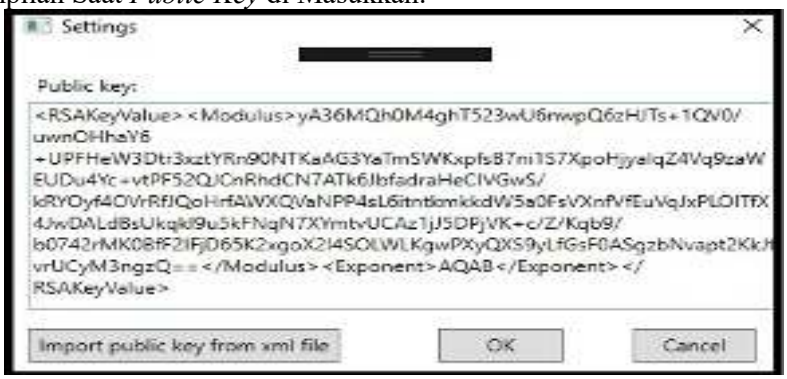
Gambar 7. Tampilan *file* Yang Sudah Dimasukkan Ke Aplikasi.

6. Tampilan *Setting* Untuk *Encrypt File.*



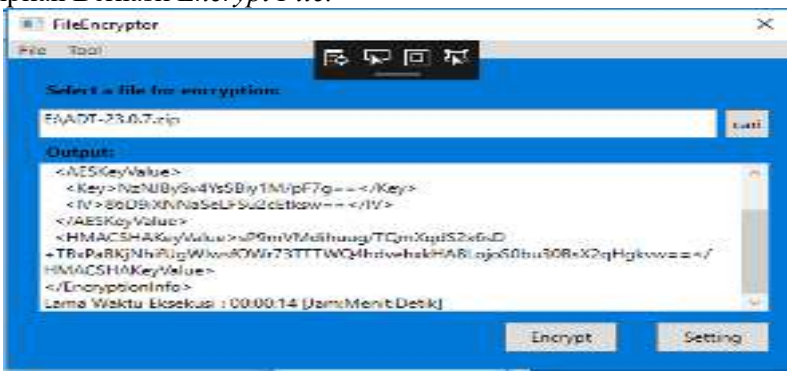
Gambar 8. Tampilan *Setting Encrypt*.

7. Tampilan Saat *Public Key* di Masukkan.



Gambar 9. Tampilan Saat *Public Key* di Masukkan.

8. Tampilan Berhasil *Encrypt File.*

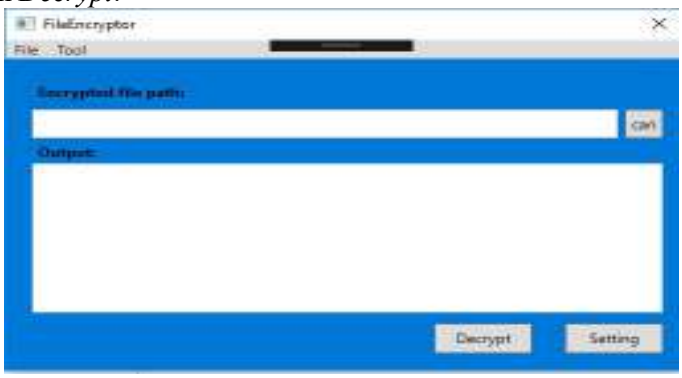


Gambar 10. Tampilan Berhasil *Encrypt File*.

9. Tampilan *Form* Menu switch.

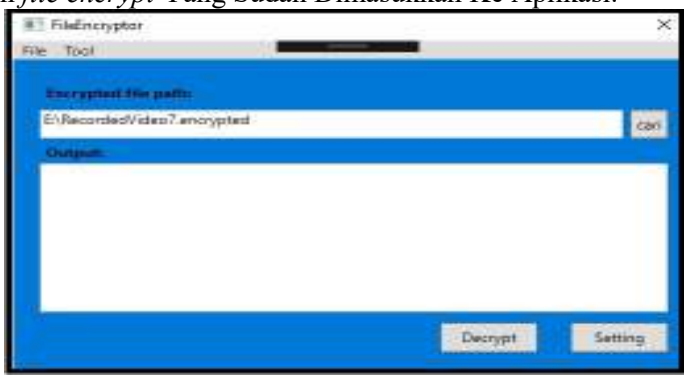
Gambar 11. Tampilan Menu switch.

10. Tampilan *Decrypt*.



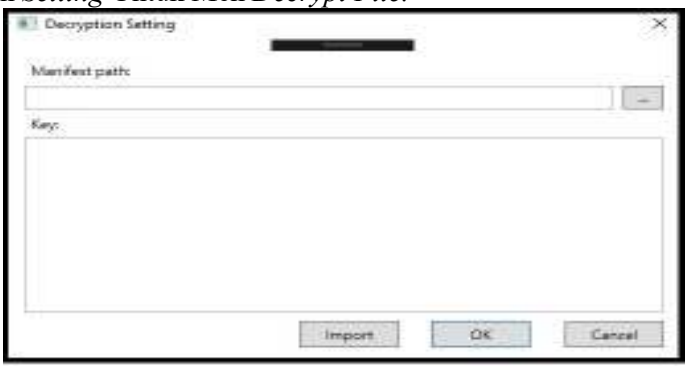
Gambar 12. Tampilan *Decrypt*.

11. Tampilan *file encrypt* Yang Sudah Dimasukkan Ke Aplikasi.



Gambar 13. Tampilan *file encrypt* Yang Sudah Dimasukkan Ke Aplikasi.

12. Tampilan *Setting* Untuk Men *Decrypt File.*

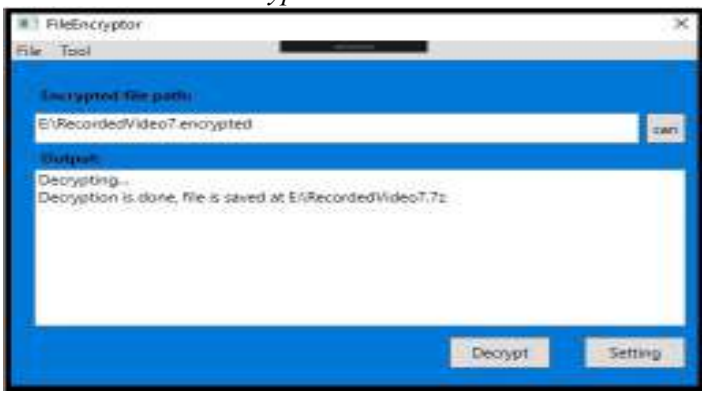


Gambar 14. Tampilan *Setting Decrypt.*

13. Tampilan Saat *Manifest path* dan *Private Key* di Masukkan.

Gambar 15. Tampilan Saat *Manifest path* dan *Private Key* di Masukkan.

14. Tampilan Berhasil Menu *Decrypt File.*



Gambar 16. Tampilan Berhasil Men *Decrypt File*.

3.4. Hasil Uji Coba.

Pengujian sistem bertujuan untuk memastikan bahwa sistem selalu tersedia. Instrumen yang digunakan untuk melakukan pengujian ini adalah melalui pengujian *black box* :

Tabel 11. *Blackbox Testing Encrypt File.*

No	*Encrypt File*	Keterangan	Hasil
1	*Klik Button* Cari untuk mencari file nya.	Sistem akan menampilkan *form* pencarian *file winrar/zip* yang akan di *encrypt*.	Valid
2	*Klik Button Setting.*	Sistem akan menampilkan *form import public key*.	Valid
3	*Klik Button Import Public Key* .	Sistem akan menampilkan kunci *publick key* nya.	Valid
4	*Klik Button Encrypt*.	Sistem akan menampilkan proses *encrypt file.*	Valid
5	Uji Kecepatan *Encrypt*	Sistem menampilkan waktu kecepatan dari *Encrypt* dengan ukuran yang berbeda	Valid

Tabel I2. *Blackbox Testing Decrypt File.*

No	*Decrypt File*	Keterangan	Hasil
1	Klik Button Cari untuk mencari file yang sudah di *encrypt*.	Sistem akan menampilkan *file* yang sudah di *encrypt* sebelumnya untuk melakukan *decrypt file*.	Valid
2	*Klik Button Setting.*	Sistem akan menampilkan *form* untuk *import manifest path* dan *import private key* yang sudah di buat sebelum nya.	Valid
3.	*Klik Button* Cari *Manifest path.*	Sistem akan menampilkan *manifest.xml* yang sudah di buat sebelum nya.	Valid
4	*Klik Button Import Private Key.*	Sistem akan menampilkan kunci *Private key* nya.	Valid

4. KESIMPULAN

Setelah melalui tahapan tahapan yang telah dijelaskan pada pembahasan sebelumnya maka dapat ditarik kesimpulan:

1. Proses perancangan untuk mengkombinasikan algoritma RSA*(Rivest Shamir Adlemen*), dan AES(*Advance Encryption Standart*) yang akan diimplementasikan pada proses pengamanan file *Winrar* & *Zip* di aplikasi *Microsoft Visual Studio* 2017 telah sesuai dengan latar belakang masalah dan batasan masalah yang ada.
2. Pengimplementasian algoritma RSA*(Rivest Shamir Adlemen)* dan AES(*Advance Encryption Standart*) dalam aplikasi *Microsoft Visual Studio* 2017 untuk mengamankan file *Winrar* & *Zip* telah berhasil dibuat.
3. Aplikasi "Aplikasi Pengamanan File *Winrar* & *Zip* Menggunakan Kombinasi Algoritma RSA*(Rivest Shamir Adlemen)*, dan AES(*Advance Encryption Standart*) berbasis *Dekstop* ini dibangun menggunakan metode pengembangan SDLC (*System Development Life Cycle*) *waterfall.*

5. SARAN

1. Saat ini aplikasi dijalankan dengan baik di sistem operasi berbasis *Windows*. Sehingga untuk keperluan pengembangan selanjutnya diharapkan dapat diimplementasikan di sistem operasi lain.
2. Pengguna disarankan untuk mengorganisir file yang dienkripsi secara bersamaan ke dalam satu folder dan juga menyertakan file secret key guna memudahkan proses dekripsi dikemudian hari.
6. Pengembangan berikutnya diharapkan dapat menambahkan fitur untuk mengirimkan file yang telah dienkripsi kepada sesorang melalui internet.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Universitas Potensi Utama yang telah membantu penulis dalam menyelesaikan laporan penelitian ini.

DAFTAR PUSTAKA

[1] Prasetyo, Y., Triandi, B., & Hardianto, H. (2018). Perancangan Aplikasi Pengamanan File Teks dengan Skema Hybrid Menggunakan Algoritma Enigma dan Algoritma RSA. *IT (INFORMATIC TECHNIQUE) JOURNAL*, *6*(1), 46-55.

[2] Tulloh, A. R., Permanasari, Y., & Harahap, E. (2016). Kriptografi Advanced Encryption Standard (AES) Untuk Penyandian File Dokumen. *Matematika*, *15*(1).

[3] Arifin, Z. (2016). Studi kasus penggunaan algoritma RSA sebagai algoritma kriptografi yang aman. *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, *4*(3), 7-14.

[4] Gunawan, I. (2018). Kombinasi algoritma Caesar cipher dan algoritma rsa untuk pengamanan file dokumen dan pesan teks. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, *2*(2), 124-129.

[5] Setyaningsih, E., & Si, S. (2015). Kriptografi & Implementasinya Menggunakan Matlab. *Yogyakarta: Andi*.