IMPLEMENTASI KEAMANAN PESAN TEKS MENGGUNAKAN KRIPTOGRAFI ALGORITMA RSA DENGAN METODE WATERFALL BERBASIS JAVA

Rudi Firmansyah[1], Angga Aditya Permana[2]
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Tangerang
Jl. Perintis Kemerdekaan I No. 33, Babakan, Kec. Tangerang, Kota Tangerang
E-mail: rudifir540@gmail.com

ABSTRAKS
Kemajuan teknologi saat ini memungkinkan pengguna bertukar data dan informasi dari satu tempat ke tempat lain melalui internet. Data dan informasi tersebut terkadang bersifat penting dan perlu dirahasiakan dari orang yang tidak bertanggung jawab yang dapat menyebar luaskan atau mengubah data dan informasi aslinya. Resiko ancaman seperti itu dapat dikurangi dengan mengubah data dan informasi ke dalam pesan rahasia. Penulis menggunakan teknik kriptografi RSA (Rivest Shamir Adleman) dalam merubah teks asli pesan menjadi teks rahasia. Penulis akan menggunakan bahasa pemrograman JAVA untuk merubah teks asli menjadi bentuk bilangan dengan blok-blok, kemudian pesan teks akan dirubah dengan kunci yang sudah dibuat terlebih dahulu. Hasil dari enkripsi pesan diharapkan mampu mengamankan data dan informasi, karena dalam kriptografi RSA belum ditemukan cara dekripsi pesan selain dengan menggunakan kunci rahasia. Metode perancangan sistem yang digunakan yaitu metode waterfall.

Kata Kunci: Kriptografi, RSA, Waterfall, Java.

ABSTRACT
Current technological advances allow users to exchange data and information from one place to another via the internet. The data and information are sometimes important and need to be kept confidential from irresponsible people who can disseminate or change the original data and information. The risk of such threats can be reduced by converting data and information into confidential messages. The author uses RSA cryptography (Rivest Shamir Adleman) technique in converting the original text of the message into secret text. The author will use the JAVA programming language to convert the original text into a number with blocks, then the text message will be changed with the key that was created first. The results of message encryption are expected to be able to secure data and information, because in RSA cryptography there has been no way to decrypt messages other than by using a secret key. The system design method used is the waterfall method.

Keywords: Cryptography, RSA, Waterfall, Java.

1. PENDAHULUAN

1.1 Latar Belakang

Teknologi komunikasi dan informasi sangat berkembang dengan pesat dan memberikan pengaruh besar bagi seluruh kehidupan manusia. Sebagai contoh perkembangan jaringan internet yang memungkinkan setiap orang untuk saling bertukar data atau informasi melalui jaringan internet tersebut. Pada proses pengiriman data (pesan) terdapat beberapa hal yang harus diperhatikan, yaitu kerahasiaan, integritas data, autentikasi dan non repudiasi. Oleh karenanya dibutuhkan suatu proses penyandian atau pengkodean pesan sebelum dilakukan proses pengiriman. Sehingga pesan yang dikirim terjaga kerahasiaannya dan tidak dapat dengan mudah diubah untuk menjaga integritas pesan tersebut.

Kriptografi (*Cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan *graphia* artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi. Contoh algoritma kriptografi yang dapat dihandalkan adalah RSA, dimana RSA merupakan proses penyandian kunci asimetrik (*asymmetric key*). Proses perumusan RSA didasarkan pada Teorema Euler, sedemikian sehingga menghasilkan kunci umum dan kunci pribadi yang saling berkaitan. Sehingga meskipun proses enkripsi dan dekripsi menggunakan dua kunci yang berbeda hasilnya akan tetap sama. Kunci umum dan kunci pribadi yang digunakan adalah suatu bilangan prima, dan disarankan bilangan prima yang besar. Hal ini digunakan untuk pencegahan usaha pemecahan teks rahasia, karena semakin besar bilangan prima yang digunakan sebagai kunci maka semakin sulit mencari bilangan besar sebagai faktornya.

Penulis akan membuat sebuah program percobaan enkripsi dan dekripsi pesan teks menggunakan teknik RSA menggunakan bahasa

pemrograman JAVA. Diharapkan pesan teks asli tidak akan berubah meskipun dalam proses enkripsi dan dekripsi menggunakan kunci yang berbeda.

1.2 Referensi

1.2.1 Kriptografi

Menurut pendapat Ginting dkk (2015) kriptografi (*cryptography*) berasal dari bahasa Yunani yang terdiri dari kata *kryptos* yang artinya tersembunyi dan *graphia* yang artinya sesuatu yang tertulis sehingga kriptografi dapat juga disebut sebagai sesuatu yang tertulis secara rahasia atau tersembunyi.

Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Kriptografi menurut Rinaldi dalam Ginting dkk (2015) juga didefinisikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek pada keamanan informasi misalnya kerahasiaan, integritas data, otentikasi pengirim/penerima data, dan otentikasi data. Dalam perkembangannya, kriptografi menurut Dony dalam Ginting dkk (2015) juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital.

Dalam kriptografi, pesan atau informasi yang dapat di baca disebut sebagai *plaintext* atau *clear text*. Proses yang dilakukan untuk mengubah teks asli (*plaintext*) ke dalam teks rahasia (*ciphertext*) disebut enkripsi. Pesan yang tidak terbaca disebut teks rahasia (*ciphertext*). Proses kebalikan dari enkripsi disebut dekripsi. Dekripsi akan mengembalikan teks rahasia (*ciphertext*) menjadi teks asli (*plaintext*). Kedua proses enkripsi dan dekripsi membutuhkan penggunaan sejumlah informasi rahasia, yang sering disebut kunci (*key*).

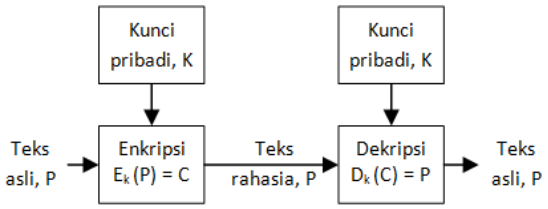
Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu:

a. Kerahasiaan (*confidentiality*), adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
b. Integritas adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain ke dalam data yang sebenarnya.
c. Autentikasi adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
d. Non-repudiasi atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/ terciptanya suatu informasi oleh yang mengirimkan/membuat.

Berbagai macam algoritma kriptografi yang terbagi menjadi dua kelompok dalam hal penggunaan kunci yaitu:

1.2.1.1 Algoritma Sandi Kunci Simetris

Skema algoritma sandi akan disebut kunci-simetris apabila untuk setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kunci yang sama. Pada gambar 2.1 dijelaskan bagaimana skema enkripsi dan dekripsi Algoritma Kriptografi Simetris.



Gambar 1.1. Algoritma Kriptografi Simetris

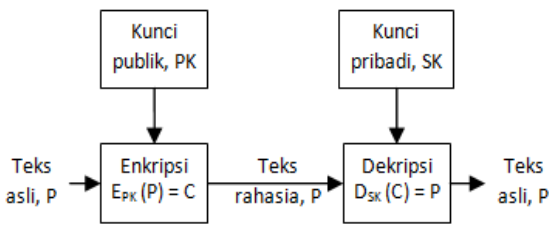
Beberapa contoh algoritma yang menggunakan kunci simetris:

a. DES - Data Encryption Standard
b. Blowfish
c. Twofish
d. MARS
e. IDEA
f. 3DES - DES diaplikasian tiga kali.
g. AES - Advanced Encryption Standard, yang bernama asli Rijndael

1.2.1.2 Algortima Sandi Kunci Asimetris

Skema ini adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Skema ini disebut juga sebagai sistem kriptografi kunci publik karena kunci untuk enkripsi dibuat untuk diketahui oleh umum (public-key) atau dapat diketahui siapa saja, tapi untuk proses dekripsinya hanya dapat dilakukan oleh yang berwenang yang memiliki kunci rahasia untuk mendekripsinya, disebut private-key.

Pada gambar 2.2 dijelaskan bagaimana skema enkripsi dan dekripsi Algoritma Kriptografi Asimetris.



Gambar 1.2. Algoritma Kriptografi Asimetris

1.2.2 Algoritma Kriptografi RSA

Menurut Ginting dkk (2015) sandi RSA merupakan algoritma kriptografi kunci publik (asimetris). Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmatika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat).

Cara menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Kenyataannya, memfaktor-kan bilangan bulat menjadi faktor primanya bukanlah pekerjaan yang mudah. Karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Cara yang bisa digunakan dalam pemfaktoran adalah dengan menggunakan pohon faktor. Jika semakin besar bilangan yang akan difaktorkan, maka semakin lama waktu yang dibutuhkan. Jadi semakin besar bilangan yang difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA.

Besaran-besaran yang digunakan pada algoritma RSA:

1. p dan q bilangan prima (rahasia)
2. $r = p * q$ (tidak rahasia) (1)
3. $\varphi(r) = (p-1)(q-1)$ (rahasia) (2)
4. PK (kunci enkripsi) (tidak rahasia)
5. SK (kunci dekripsi) (rahasia)
   $SK * PK = 1(mod * \varphi(r))$ (3)
6. X (plaintext) (rahasia)
7. Y (ciphertext) (tidak rahasia)

1.2.2.1 ASCII System

Plain teks yang akan dienkripsi dengan RSA *Coding* merupakan angka-angka, sedangkan pesan yang dikirimkan biasanya berbentuk teks atau tulisan. Sehingga dibutuhkan suatu kode yang sifatnya *universal* untuk mengubah pesan teks menjadi *plain* teks dalam bentuk bilangan. ASCII (American Standard Code for Information Interchange) atau Kode Standar Amerika untuk pertukaran informasi merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks.

1.2.2.2 Aritmatika Modulo

Dalam penerapan Teorema Euler pada perumusan algoritma RSA *Coding* sangat dibutuhkan pemahaman tentang modulo. Modulo sendiri berarti sisa hasil bagi. Misalkan a adalah bilangan bulat dan m adalah bilangan bulat dimana a dan m lebih besar dari 0. Maka operasi a mod m (dibaca "a modulo m") memberikan sisa jika a dibagi dengan m. Bilangan m disebut modulus atau modulo, dan hasil modulo m terletak di dalam himpunan {0, 1, 2, …, m–1}.

Contoh : Diambil a = 20 dan m = 6 . Karena 20 dibagi 6 adalah 3 bersisa 2, maka diperoleh a mod m ≡ 20 mod 6 ≡ 2.

1.2.2.3 Pembangkitan Pasangan Kunci

Sebagai algoritma Asimetris Kriptografi, pengkodean RSA membutuhkan dua kunci yang berbeda untuk enkripsi dan dekripsi. Bilangan yang dipilih sebagai kunci adalah bilangan prima yang besar, dengan alasan pemfaktoran sebuah bilangan hasil perkalian dari dua bilangan prima yang besar menjadi dua bilangan prima yang sesuai akan sangat sulit. Sehingga keamanan dari RSA *Coding* dapat terjamin. Berikut langkah-langkah proses pembangkitan pasangan kunci pada RSA:

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung r dengan persamaan (1). Sebaiknya $p \neq q$, sebab jika p = q maka r = p2 sehingga p dapat diperoleh dengan menarik akar pangkat dua dari r.
3. Hitung φ(r) menggunakan persamaan (2).
4. Pilih kunci publik, PK, yang relatif prima terhadap φ(r).
5. Bangkitkan kunci rahasia dengan menggunakan persamaan (3).

Perhatikan bahwa SK · PK = 1 (mod φ(r)) ekuivalen dengan SK · PK = 1 + mφ(r), sehingga SK dapat dihitung dengan persamaan (4).

$$SK = \frac{1+m\varphi(r)}{PK} \quad (4)$$

1.2.2.4 Proses Enkripsi

Langkah-langkah pada proses enkripsi adalah sebagai berikut :

1. *Plaintext* diubah ke dalam bentuk bilangan. Untuk mengubah *plaintext* yang berupa huruf menjadi bilangan dapat digunakan kode ASCII dalam sistem bilangan decimal.
2. *Plaintext* m dinyatakan menjadi blok-blok x1,x2,x3,...., sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n-1] , sehingga transformasinya menjadi satu ke satu.
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus
   $y_i = {x_i}^{PK} \; mod \; r$ (5)

1.2.2.5 Proses Dekripsi

Langkah-langkah pada proses dekripsi adalah sebagai berikut :

1. Setiap blok *ciphertext* y_i didekripsi kembali menjadi blok x_i dengan rumus

$$x_i = {y_i}^{SK} \; mod \; r \qquad (6)$$

2. Kemudian blok-blok m1,m2,m3,...., diubah kembali ke bentuk huruf dengan melihat kode ASCII hasil dekripsi.

1.2.2.6 Kekuatan dan Keamanan RSA

Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini r = p*q. Sekali r berhasil difaktorkan menjadi p dan q, maka φ(r) = (p – 1) (q – 1) dapat dihitung. Selanjutnya, karena kunci enkripsi PK diumumkan (tidak rahasia), maka kunci dekripsi SK dapat dihitung dari persamaan (3). Penemu algoritma RSA menyarankan nilai p dan q panjangnya lebih dari 100 digit. Dengan demikian hasil kali r = p*q akan berukuran lebih dari 200 digit. Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik).

Untunglah algoritma yang paling mangkus untuk memfaktorkan bilangan yang besar belum ditemukan. Inilah yang membuat algoritma RSA tetap dipakai hingga saat ini. Selagi belum ditemukan algoritma yang mangkus untuk memfaktorkan bilangan bulat menjadi faktor primanya, maka algoritma RSA tetap direkomendasikan untuk menyandikan pesan.

1.2.2.7 Contoh Perhitungan

Pilih nilai p dan q dengan syarat:

$$\left| \begin{array}{c} p \;\&\; q = bilangan\ prima \\ p \neq q \\ p = 11, \quad q = 13 \end{array} \right|$$

Hitung nilai r

$$\left| \begin{array}{l} r = p \cdot q \\ \quad = 11 \cdot 13 \\ \quad = 143 \end{array} \right|$$

Hitung nilai φ(r)

$$\left| \begin{array}{l} \varphi(r) = (p-1) \cdot (q-1) \\ \quad = (11-1) \cdot (13-1) \\ \quad = 120 \end{array} \right|$$

Mencari nilai kunci PK (kunci publik)

$$\left| \begin{array}{c} 1 < PK < \varphi(r) \\ GCD\big(PK, \varphi(r)\big) = 1 \\ dipilih\ PK = 59 \end{array} \right|$$

Hitung nilai SK (kunci rahasia)

$$\left| \begin{array}{c} SK = \dfrac{1 + m \cdot \varphi(r)}{PK} \\ dengan\ mencoba\ m = 1, m = 2, dst \\ dengan\ hasil\ bilangan\ bulat \\ maka\ diperoleh\ m = 29 \\ SK = \dfrac{1 + m \cdot \varphi(r)}{PK} \\ = \dfrac{1 + (29 \cdot 120)}{59} \\ = 59 \end{array} \right|$$

Maka didapat kunci PK = 59 dan SK = 59.

Enkripsi pesan teks kata “Tes” dengan terlebih dahulu mengubah teks menjadi ASCII desimal (tabel 1.1).

Tabel 1.1. Konversi Teks menjadi Bilangan

Karakter	T	e	s
ASCII des	84	101	115

Enkripsi pesan x = 84101115

$$\left| \begin{array}{c} y_1 = {x_1}^{PK} mod\ r = 84^{59} mod\ 143 = 63 \\ y_2 = {x_2}^{PK} mod\ r = 101^{59} mod\ 143 = 17 \\ y_3 = {x_3}^{PK} mod\ r = 115^{59} mod\ 143 = 97 \\ y = 631797\ (chipertext) \end{array} \right|$$

Dekripsi pesan y = 631797

$$\left| \begin{array}{c} x_1 = {y_1}^{SK} mod\ r = 63^{59} mod\ 143 = 84 \\ x_2 = {y_2}^{SK} mod\ r = 17^{59} mod\ 143 = 101 \\ x_3 = {y_3}^{SK} mod\ r = 97^{59} mod\ 143 = 115 \\ x = 84101115\ (pesan\ asli) \end{array} \right|$$

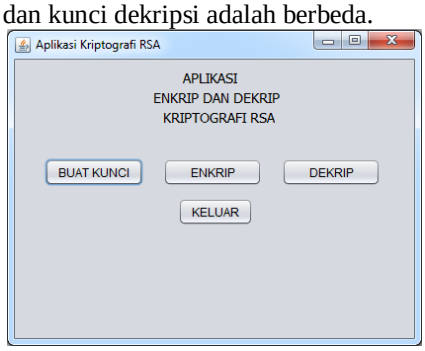
Konversikan kembali x menggunakan tabel ASCII (tabel 1.2).

Tabel 1.2. Konversi Bilangan menjadi Teks

ASCII des	84	101	115
Karakter	T	e	s

2. PEMBAHASAN

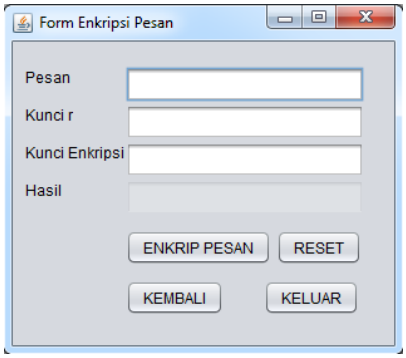
Rancangan *user interface* program aplikasi enkripsi dan dekripsi algoritma RSA digambarkan pada Gambar 2.1 sampai 2.4. Pada Gambar 2.2 terdapat form untuk membuat kunci RSA secara otomatis, karena pada kriptografi RSA kunci enkripsi dan kunci dekripsi adalah berbeda.



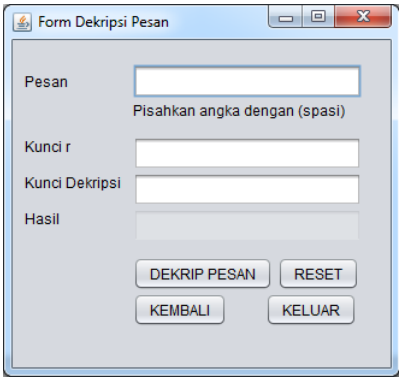
Gambar 2.1. UI Form Utama

Gambar 2.2. UI Form Buat Kunci

ISSN: 2503-071X
E-ISSN: 2621-511X

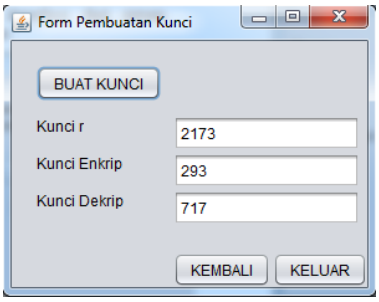


Gambar 2.3. UI Form Enkripsi



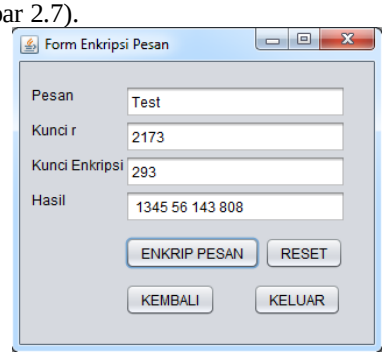
Gambar 2.4. UI Form Dekripsi

Pengujian sistem dilakukan dengan langkah pembuatan kunci pada form buat kunci dengan hasil seperti pada Gambar 2.5.

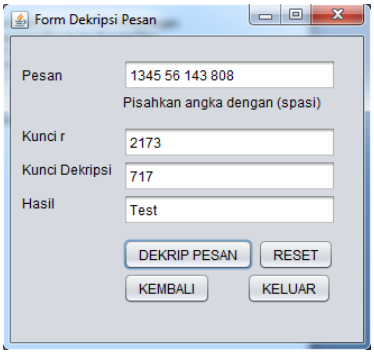


Gambar 2.5. Hasil Pemrosesan pada Form Buat Kunci

Pembuktian bahwa kunci yang dihasilkan pada form buat kunci dengan memasukkan kunci r=2173, kunci enkripsi=293, dan kunci dekripsi=717 pada form enkripsi dan form dekripsi (lihat Gambar 2.6 dan Gambar 2.7).



Gambar 2.6. Proses Enkripsi



Gambar 2.7. Proses Dekripsi

Pada sistem terlihat bahwa dengan kunci yang dibuat secara otomatis dengan form buat kunci berhasil merubah teks asli "Test" menjadi teks rahasia "1345 56 143 808" dan dikembalikan lagi menjadi teks asli "Test".

3. KESIMPULAN DAN SARAN

Berdasarkan pengujian yang dilakukan penulis dapat disimpulkan bahwa kriptografi RSA ini cukup aman, karena kunci RSA tidak asal dibuat. Hasil dari kriptografi ini juga cukup membingungkan bagi pembaca yang tidak mempunyai kunci dekripsi, karena yang dihasilkan hanya berupa bilangan buat.

Kedepannya penulis berharap aplikasi ini dapat dikembangkan menjadi enkripsi dan dekripsi file.

PUSTAKA

Ginting, A., Isnanto, R.R., & Windasari, I.P. 2015. *Implementasi Algoritma Kriptografi Rsa untuk Dekripsi dan Enkripsi Email*, Jurnal Teknologi dan Sistem Komputer, Vol. 3, No. 2, April 2015, e-ISSN:2338-0403.

Pressman, R.S. 2002. *Rekayasa Perangkat Lunak: Pendekatan Praktisi(Buku Dua).* Yogyakarta: Penerbit Andi.

Santoso, H. & Fakhriza, M. 2018. *Perancangan Aplikasi Keamanan File Audio Format Wav (Waveform) Menggunakan Algoritma Rsa.* Jurnal Ilmu Komputer dan Informatika, Vol. 2, No. 1, April 2018, ISSN 2598-6341.

Sasmito, G.W. 2017. *Penerapan Metode Waterfall pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal, Jurnal Informatika: Jurnal Pengembangan IT (JPIT).* Vol. 2, No. 1, Januari 2017, ISSN: 2477-5126 dan e-ISSN: 2548-9356.

Sommerville, I. 2011. *Software Engineering 9th Edition.* Addison-Wesley.

Warno. 2012. *Pembelajaran Pemrograman Bahasa Java dan Arti Keyword.* Jurnal Komputer Volume 8, Nomor 1, Maret 2012