Seminar Nasional Ilmu Komputer (SNIK 2016) - Semarang, 10 Oktober 2016
ISBN: 978-602-1034-40-8

Mengukur Kecepatan Enkripsi dan Dekripsi Algoritma RSA pada Pengembangan Sistem Informasi *Text Security*

Desi Wulansari[1], Alamsyah[2], Fajar Arif Setyawan[3], Hendi Susanto[4]

[1,2,3,4]Jurusan Ilmu Komputer, FMIPA, Universitas Negeri Semarang
Email: [1] desiwulans03@gmail.com , [2]alamsyah@mail.unnes.ac.id, [3]fajarmath@gmail.com, [4]hendi.susanto@outlook.com

Abstrak

Keamanan dan kerahasiaan data merupakan hal penting untuk mencegah penyalahgunaan data. Sebuah data dapat diamankan dengan cara melakukan penyandian terhadap isi data. Salah satu metode dalam penyandian data yang masih populer adalah kriptografi RSA. Objek penelitian ini adalah proses implementasi algoritma kriptografi RSA pada nilai parameter *n* dengan ukuran 1024 bit dan 2048 bit. Proses yang diamati adalah kompleksitas waktu yang dihasilkan oleh instruksi enkripsi dan dekripsi. Tahap-tahap yang dilakukan adalah studi pendahuluan, mengumpulkan data, menganalisis kebutuhan, pengembangan dan pengujian sistem informasi serta penarikan kesimpulan. Hasil pengujian menyatakan algoritma RSA 1024 bit memiliki rata-rata kecepatan enkripsi sebesar 352.488 *nano second* dan rata-rata kecepatan dekripsi sebesar 109.347.917 *nano second*, sedangkan pada algoritma RSA 2048 bit memiliki rata-rata kecepatan enkripsi sebesar 1.772.900 *nano second* dan rata-rata kecepatan dekripsi sebesar 775.282.334 *nano second*.
Kata Kunci: Algoritma RSA, Enkripsi, Dekripsi

Abstract

Security and confidentiality of data is essential to prevent the misuse of data. A data can be secured using encryption of the data contents. A method of encrypting data that is still popular is the RSA *cryptography. The object of this research is the process of implementation of* RSA *cryptography algorithm on the n value parameter with the size of* 1024 bit *and* 2048 bit. *The process observed is the time complexity generated by the instruction of encryption and decryption. The stages are performed is a preliminary study, collecting data, analyzing the needs, development and testing of information system as well as the conclusion. The test results stated a* 1024 bit RSA *algorithm has an average speed of* 352.488 *nano seconds encryption and decryption average speed of* 109.347.917 *nano second, while in the* 2048 bit RSA *algorithm has an average speed* 1.772.900 *nano seconds of encryption and average* 775.282.334 *nano seconds of decryption.*
Keyword: RSA *Algorithm, Encryption, Decryption*

1. PENDAHULUAN

Saat ini, kemajuan proses transfer informasi menjadi suatu keniscayaan. Perkembangan teknologi informasi yang pesat menyebabkan akses informasi yang mudah dan cepat. Kemudahan dalam mengakses informasi inilah yang kemudian menuntut pemilik informasi untuk dapat menjaga data-data atau informasi penting dan rahasia agar tidak mudah diketahui oleh pihak lain yang tidak memiliki kewenangan sehingga dapat mencegah terjadinya penyalahgunaan informasi atau data-data tersebut. Sebuah data dapat diamankan dengan cara melakukan penyandian terhadap isi data dan hal ini biasa disebut dengan kriptografi.

Kriptografi berasal dari bahasa Yunani yaitu *kripto* (tersembunyi) dan *grafh* (tulisan) yang artinya tulisan yang tersembunyi [1]. Kriptografi secara umum merupakan ilmu dan seni untuk menjaga keamanan pesan [2]. Secara umum, kriptografi terdiri atas dua buah yaitu kriptografi kunci publik dan kriptografi kunci privat. Pada sistem kriptografi kunci public, kunci untuk enkripsi sama dengan kunci untuk dekripsi [3]. Salah satu algortima kriptografi kunci publik yang terkenal adalah kriptografi RSA. Dalam segi keamanan algoritma RSA dinilai memiliki keamanan yang baik serta masih menjadi algoritma kriptogafi kunci publik yang populer dan banyak digunakan [4]. Operasi matematika pada RSA terdiri dari perkalian dua buah bilangan prima (*p* dan *q*) yang dipilih secara acak, hasil dari perkalian tersebut adalah *n*. Semakin besar ukuran *n*, semakin bagus tingkat keamanannya karena akan menyulitkan penyerang untuk memecahkan faktorisasi dari nilai *n* [1]. Kemudian dilanjutkan dengan proses operasi perpangkatan dan operasi modular. Perhitungan modular pada proses pembagian harus menyisakan hasil bagi (nilai *remainder*).

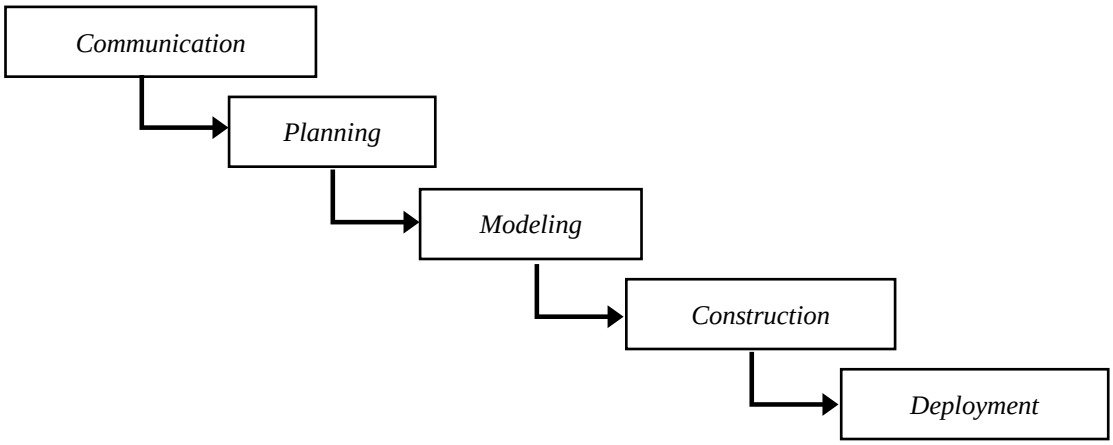
Pertanyaan yang muncul adalah mengenai lamanya waktu yang dibutuhkan algoritma RSA untuk melakukan proses enkripsi dan dekripsi jika diasumsikan bahwa nilai *n* bit-nya berukuran 1024 bit dan

ISBN: 978-602-1034-40-8

2048 bit. Oleh karena itu, perlu dilakukan kajian mengenai implementasi algoritma RSA pada nilai *n* yang berukuran 1024 bit dan 2048 bit melalui pengembangan suatu sistem informasi *Text security* dalam bahasa pemrograman Java.

2. METODE

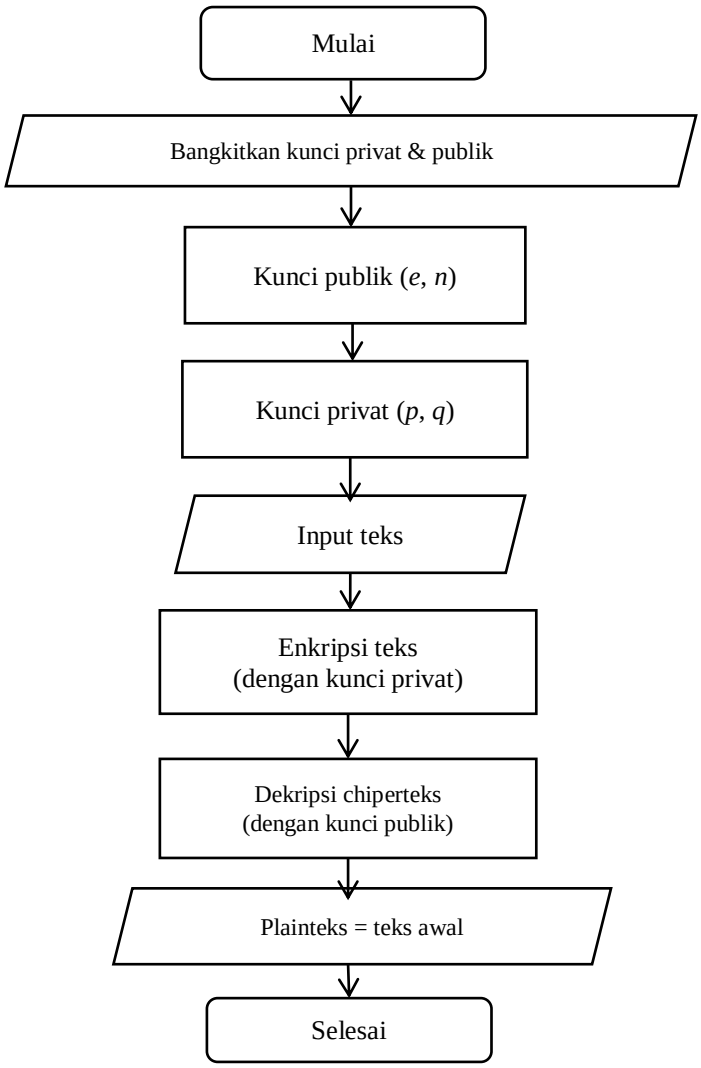
Objek penelitian ini adalah proses implementasi algoritma kriptografi RSA pada nilai parameter *n* dengan ukuran sebesar 1024 bit dan 2048 bit. Proses yang diamati adalah kompleksitas waktu yang dihasilkan oleh instruksi enkripsi dan dekripsi pada algoritma kriptografi RSA. Menurut Burch dan Grudnitski, kualitas informasi ditentukan oleh tiga faktor yaitu relevansi, ketepatan waktu dan akurasi [5]. Variabel yang diteliti terdiri dari 2 kriteria, yaitu kecepatan algoritma dalam memproses perintah enkripsi dan dekripsi terhadap ukuran parameter *n* sebesar 1024 bit dan 2048 bit pada algoritma RSA. Dua variabel ini diterapkan pada data berupa teks sebagai bahan uji selama proses simulasi. Perangkat lunak yang digunakan untuk membangun sistem adalah *Eclipse Mars* dan *Time-Stamp Counter*. Pengujian sistem dilakukan pada laptop dengan spesifikasi CPU AMD C-60 APU *with* Radeon™ HD *Graphics* 1.00 GHz, RAM 2.00 GB dan Windows 7 *Operating System*. Sistem informasi dirancang menggunakan model *waterfall* yang bersifat klasik, sistematis dan berurutan dalam membangun *software* [6]. Berikut ini merupakan fase-fase dalam model *waterfall* menurut Pressman yang ditunjukkan pada Gambar 1.



Gambar 1. Model *waterfall*.

Bahasa pemrograman yang digunakan untuk membangun sistem implementasi algoritma kriptografi RSA dengan ukuran *n* sebesar 1024 bit dan 2048 bit adalah bahasa pemrograman Java. Sebelum membangun sistem, terlebih dahulu menentukan *flowchart* dari cara kerja algoritma RSA seperti yang ditampilkan pada Gambar 2.

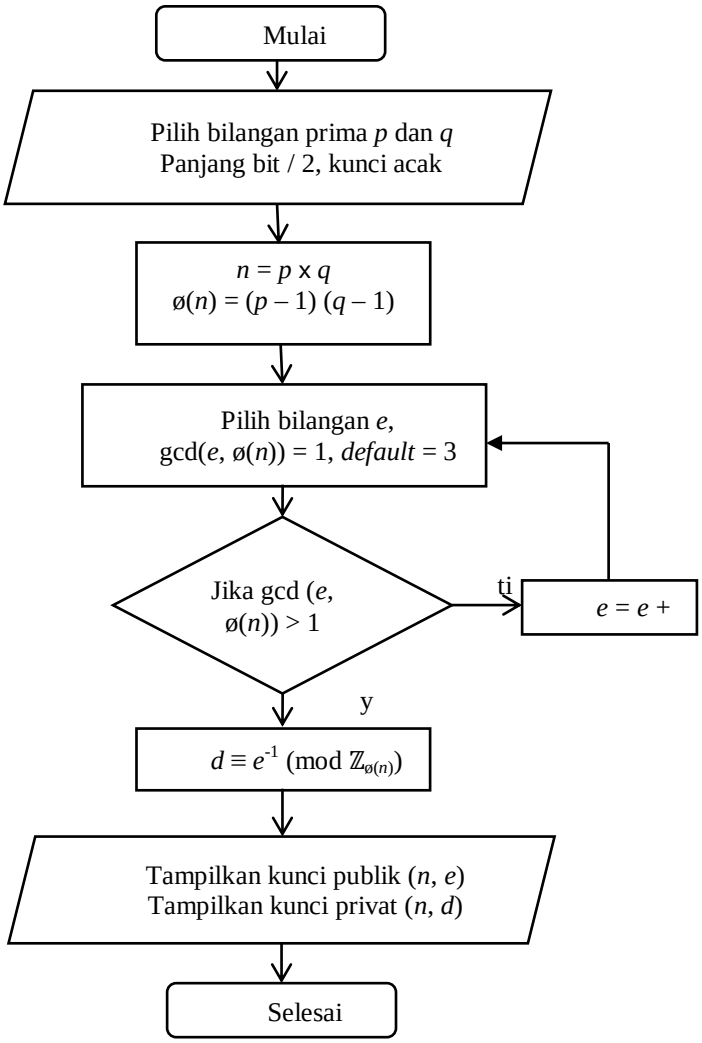
ISBN: 978-602-1034-40-8



Gambar 2. *Flowchart* algoritma kriptografi RSA.

Adapun *flowchart* untuk pembangkit kunci RSA pada sistem dijelaskan pada Gambar 3.

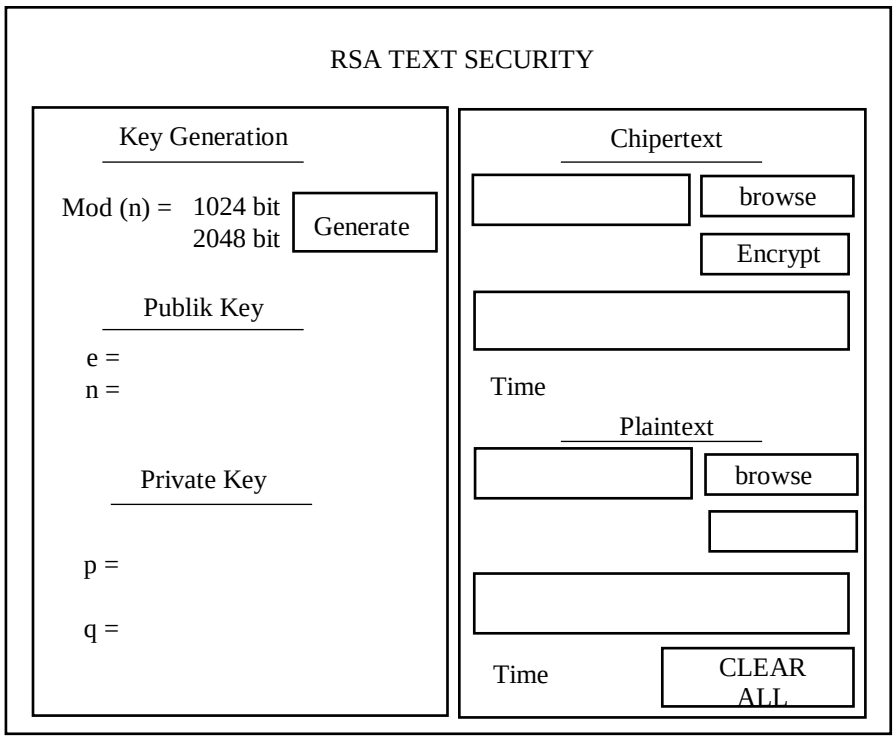
ISBN: 978-602-1034-40-8



Gambar 3. *Flowchart* pembangkit kunci kriptografi RSA.

Time Stamp Counter (TSC) digunakan untuk mengetahui kompleksitas waktu algoritma kriptografi RSA dengan ukuran *n* sebesar 1024 bit dan 2048 bit melalui instruksi *read-time stamp counter* (RDTSC). Instruksi tersebut merupakan proses untuk mencatat waktu atau *clock cycle* yang dihasilkan di bawah kendali administrator. Proses ini mencatat waktu saat perintah enkripsi dieksekusi hingga selesai, begitu juga dengan perintah dekripsi. Setelah sistem dibangun maka berikutnya adalah melakukan pengujian sistem untuk memperoleh hasil berupa waktu yang dibutuhkan algoritma RSA dalam melakukan proses enkripsi dan dekripsi. Pada pilihan modulus (*n*), pengguna dapat memilih salah satu besaran bit kemudian selanjutnya kunci akan dibangkitkan setelah tombol *generate* ditekan. Pengujian menggunakan 5 buah data teks, yang masing-masing datanya akan diuji sebanyak 5 kali. Berikut adalah rancangan *interface* sistem RSA ditunjukan pada Gambar 4.

ISBN: 978-602-1034-40-8



Gambar 4. Rancangan *Interface* Sistem RSA

3. HASIL DAN PEMBAHASAN

Keutamaan dari kriptografi RSA adalah tantangan pada ukuran modulus *n* yang dimliki RSA [7], algoritma ini dikatakan aman jika penyerang sulit dalam memfaktorkan modulus *n* menjadi *p* dan *q* [8][9]. Penelitian ini menghasilkan sebuah sistem kriptografi dengan nama "RSA *Text Security*". Adapun data yang digunakan sebagai bahan uji berupa plainteks sebanyak 5 buah dengan masing-masing data memiliki ukuran dan karakteristik yang berbeda. Pada data teks pertama plainteks yang digunakan adalah huruf *lower case*. Data teks kedua menggunakan huruf *upper case*. Data teks ketiga menggunakan kombinasi dari *upper case*, *lower case* dan angka. Data keempat meggunakan kombinasi antara *upper case*, *lower case*, angka dan simbol. Sedangkan pada data kelima menggunakan kombinasi antara *upper case*, *lower case*, dan simbol. Proses pengujian dilakukan dengan cara setiap data teks diuji sebanyak 5 kali pada proses enkripsi dan dekripsi. Hal ini bertujuan untuk mendapatkan waktu tercepat dari proses enkripsi dan dekripsi pada tiap data uji. Setelah itu, hasil tecepat akan dicatat dan dibandingkan dengan hasil dari data uji yang lain.

Pada pengujian pertama, kunci yang dibangkitkan sebesar 1024 bit, lalu langkah selanjutnya adalah menekan tombol *generate* untuk memulai proses pembangkitan kunci. Setelah proses pembangkitan kunci selesai, hasil kunci yang diperoleh secara *random* ditampilkan pada kolom *Public Key* dan *Private Key*. Nilai *e* yang dihasilkan oleh sistem secara *random* sesuai dengan ketentuan e ($1<e<$ ø(n)) dengan gcd (e, ø(n)) = 1 adalah e = 3. Karena ukuran kunci sebesar 1024 bit maka jumlah bilangan yang dibangkitkan cukup panjang pada nilai *p*, *q* dan *n*. Nilai kunci *p* dibangkitkan sebanyak 154 karakter dalam bilangan desimal. Nilai kunci *q* yang dibangkitkan jumlahnya tidak berbeda dengan nilai kunci *p* yaitu sebanyak 154 karakter. Selanjutnya sistem memproses langkah untuk mencari nilai dari perkalian kedua bilangan prima *p* dan bilangan prima *q* tersebut dengan mengalikan hasil dari keduanya yaitu $n = p \times q$. Nilai kunci yang dihasilkan melalui proses pembangkitan kunci memiliki jumlah karakter sebanyak 308 bilangan desimal. Hal ini memungkinkan RSA dapat memenuhi keamanan karena jumlah digit *p* dan *q* lebih dari 200 karakter sehingga menyulitkan penyerang dalam melakukan faktorisasi bilangan bulat pada variabel *n* [10]. Jumlah karakter kunci *n* yaitu jumlah digit kunci *p* ditambah jumlah digit kunci *q*. Total ukuran dari masing-masing nilai kunci *p* dan *q* adalah sebesar 512 bit. Tahap selanjutnya adalah mengubah pesan asli menjadi chiperteks melalui proses enkripsi dengan menggunakan kunci publik (*n*, *e*). Dalam proses pengujian, ada 5 buah teks yang digunakan dengan masing-masing memiliki ukuran berbeda. Pengujian ini dilakukan untuk mengetahui kompleksitas waktu algoritma RSA jika melakukan proses enkripsi dan dekripsi pada ukuran *n* sebesar 1024 bit.

ISBN: 978-602-1034-40-8

Hasil pengujian proses enkripsi dan dekripsi pada ke 5 data teks yang diterapkan pada algoritma RSA dengan ukuran *n* 1024 bit yang diambil berdasarkan waktu tercepat setelah melalui 5 kali proses pengujian pada tiap data teks disajikan pada Tabel 1.

Tabel 1. Hasil pengujian kompleksitas waktu algoritma RSA 1024 bit

Data Teks ke-	Ukuran (byte)	Waktu Enkripsi (*nano second*)	Waktu Dekripsi (*nano second*)
1	3 byte	359.870	108.506.579
2	5 b0 byte	371.148	108.708.557
3	15 byte	338.340	106.816.931
4	15 byte	336.289	117.587.412
5	40 byte	356.795	105.120.106

Langkah awal pengujian kriptografi RSA pada ukuran *n* 2048 bit sama dengan langkah pengujian sebelumnya. Yaitu kunci pada RSA 2048 bit dibangkitkan terlebih dahulu. Jumlah karakter pada kunci *p* yang berhasil dibangkitkan pada RSA 2048 bit adalah sebanyak 309 bilangan desimal, sedangkan pada kunci *q* jumlah karakter kunci sebanyak 309 bilangan desimal juga. Masing-masing ukuran bit dari bilangan prima *p* dan bilangan prima *q* sebesar 1024 bit. Jumlah karakter kunci *n* yang diperoleh melalui hasil perkalian antara bilangan prima *p* dan bilangan prima *q* sebanyak 617 bilangan desimal. Kunci ini dianggap memenuhi kriteria keamanan karena jumlah karakter yag dibangkitkan lebih dari 200 bilangan desimal. Kekurangannya adalah pada proses pembangkit kunci dengan ukuran *n* sebesar 2048 bit membutuhkan waktu yang lebih lama jika dibandingkan dengan proses pembangkit kunci dengan ukuran *n* sebesar 1024 bit. Setelah proses pembangkitan kunci pada RSA selesai, proses pengujian selanjutnya adalah proses menyandikan pesan (enkripsi). Hasil secara keseluruhan berupa kompleksitas waktu RSA 2048 bit dalam memproses enkripsi dan dekripsi yang diterapkan pada ke 5 data teks ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Kompleksitas Waktu Algoritma RSA 2048 bit

Data Teks ke-	Ukuran (*byte*)	Waktu Enkripsi (*nano second*)	Waktu Dekripsi (*nano second*)
1	3 byte	1.760.392	777.992.743
2	50 byte	1.808.580	781.631.433
3	15 byte	1.769.619	787.128.942
4	15 byte	1.750.139	765.251.689
5	40 byte	1.775.771	764.406.865

Berdasarkan pengujian yang diterapkan pada 5 data teks dan menempuh 5 kali pengujian pada masing-masing data teks, diperoleh hasil bahwa pada proses dekripsi pesan membutuhkan waktu yang cukup lama jika dibandingkan dengan proses enkripsi.

4. SIMPULAN

Hasil dari pengujian menyatakan bahwa algoritma RSA memiliki rata-rata kecepatan sebesar 109.347.917 *nano second* dalam melakukan proses dekripsi dan rata-rata kecepatan sebesar 352.488 *nano second* dalam melakukan proses enkripsi. Pada pengujian algoritma RSA 2048 bit, proses dekripsi memiliki rata-rata kecepatan sebesar 775.282.334 *nano second* dan rata-rata kecepatan enkripsi sebesar 1.772.900 *nano second*. Terdapat kekurangan pada proses pembangkitan kunci RSA 2048 bit karena waktu yang dibutuhkan lebih lama. Kemudian, penerapan algoritma RSA pada *source code* Java cocok menggunakan *class BigInteger* karena dapat membangkitkan bilangan dengan jumlah besar dalam proses pembangkitan kunci 1024 bit dan 2048 bit.

5. REFERENSI

[1] Rifki Sadikin. 2012. *Kriptografi untuk Keamanan Jaringan*. Edisi Pertama. Penerbit ANDI, Yogyakarta.
[2] Chen, C., Wang, T., & Tian, J. 2013. Improving Timing Attack on RSA-CRT via Error Detection and Correction Strategy. *Information Sciences*. Vol.232(2013), 464-474.
[3] Rinaldi, Munir. 2006. *Kriptografi*. Bandung: Informatika Bandung.
[4] Bruce, S., Kathleen, S. and Saranya, V. 2016. *Worldwide survey of Encryption Products*. Harvard University, USA.
[5] Abdul, Kadir. 2003. *Pengenalan Sistem Informasi*. ANDI Yogyakarta, Yogyakarta.

ISBN: 978-602-1034-40-8

[6] Roger S. Pressman. 2010. Software Engineering: *A Practitioner's Approach*. McGraw-Hill, New York.
[7] William Stalling. 2013. *Cryptography and Network Security: Principles and Practice, Sixth Edition*. Pearson, Upper Saddle River.
[8] Rivest, R. L., Shamir, A., & Adleman, L. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*. Vol. 21(2): 120-126.
[9] Verma, S., & Garg, D. (2014). An Improved RSA Variant. *International Journal of Advancements in Technology*. Vol.5(2): 161-169.
[10] Sing,Y. Yan. (2008). *Cryptanalytic Attacks on RSA*. Springer, New York.