MATHEMATICAL SCIENCES AND APPLICATIONS JOURNAL

Volume 4, Number 2 | ISSN: 2986-4180

https://online-journal.unja.ac.id/msa

Article

Implementasi Algoritma RSA (Rivest-Shamir-Adleman) pada Kriptografi Klasik

Muhammad Zaky Zachary, Sisilia Sylviani*, Edi Kurniadi

Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran, Sumedang 45363, Indonesia

*Corresponding author e-mail: sisilia.sylviani@unpad.ac.id

Article Info
Received October 2023
Accepted October 2023
Published April 2024

Keyword:
RSA algorithm
Classical cryptography
Encryption
Decryption

Abstract

The implementation of the RSA (Rivest-Shamir-Adleman) cryptography algorithm, which is one of the most commonly used public-key cryptography algorithms. RSA provides high security and is widely used in information security applications such as data encryption, digital signature generation, and key exchange. This research documents the step-by-step implementation process of the RSA algorithm in classical cryptography, including key generation, data encryption, and decryption, leading to the generation of encryption and decryption tables for a specific key.

1. Pendahuluan

Algoritma RSA pertama kali diperkenalkan pada tahun 1976 oleh tiga peneliti yang berasal dari Massachusetts Institute of Technology, yaitu Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari inisial ketiga peneliti tersebut[1, 2] . Algoritma ini berdasarkan konsep bilangan prima dan aritmatika modulo dalam proses enkripsi dan dekripsi, dengan kedua kunci, baik kunci enkripsi maupun kunci dekripsi, merupakan bilangan bulat. Kunci enkripsi dapat diketahui oleh publik, tetapi kunci dekripsi harus dirahasiakan. Kunci dekripsi dibuat dengan menggabungkan beberapa bilangan prima bersama dengan kunci enkripsi. Untuk menemukan kunci enkripsi, seseorang harus melakukan faktorisasi pada suatu bilangan non-prima hingga ditemukan faktor-faktor prima yang membentuknya. Nyatanya, melakukan faktorisasi pada bilangan non-prima menjadi faktor-faktor primanya adalah tugas yang sulit. Hingga saat ini belum ditemuka algoritma yang efisien untuk melakukan faktorisasi tersebut. Semakin besar bilangan non-prima tersebut, semakin sulit menemukan faktor-faktor primanya. Tingkat kesulitan dalam faktorisasi ini berkontribusi langsung pada kekuatan algoritma RSA [3–5].

Kriptografi adalah ilmu untuk menjaga rahasia tetap rahasia. Misalkan pengirim yang disebut di sini dan pada tulisan berikutnya sebagai “Alice” (sebagaimana umumnya digunakan) ingin mengirim pesan *m* kepada seorang penerima yang disebut sebagai “Bob”. Dia menggunakan saluran komunikasi yang tidak aman, contohnya yaitu saluran tersebut bisa berupa jaringan komputer atau jalur telepon. Masalah muncul jika pesan tersebut mengandung informasi rahasia. Pesan tersebut bisa disadap dan dibaca oleh penyadap. Bahkan yang lebih buruk lagi adalah lawan yang biasanya disebut sebagai “Eve”, mungkin dapat memodifikasi pesan selama transmisi sedemikian rupa sehingga penerima sah, yaitu “Bob” tidak mendeteksi manipulasi tersebut [6, 7].

Kriptografi klasik adalah teknik kriptografi yang digunakan sebelum adanya komputer atau teknologi modern [8, 9]. Teknik ini melibatkan pengacakan huruf pada kata terang atau *plaintext* menggunakan penggantian huruf atau substitusi dan pengacakan posisi huruf atau transposisi [10].

Sifat pembagian yang terdapat pada bilangan bulat menghasilkan konsep seperti bilangan prima dan konsep aritmetika modulo. Aritmetika modulo berfokus pada peran penting dalam komputasi bilangan bulat khususnya dalam pengaplikasiannya pada kriptografi. Operator pada aritmetika modulo adalah **mod** yang memberikan sisa pembagian [11].

https://doi.org/ 10.22437/msa.v4i2.28863

Bilangan prima adalah bilangan yang hanya bisa dibagi oleh bilangan itu sendiri dan 1 [12–15]. Sebagai contoh, angka 8 tidak termasuk bilangan prima karena bisa dibagi oleh 2 dan 4. Bilangan-bilangan prima yang umumnya dikenal adalah 2, 3, 5, 7, 11, 13, dan seterusnya, dan daftar ini terus berlanjut. Banyaknya bilangan prima adalah tak terbatas. Meskipun kita terus mencari dalam jumlah yang besar, kita akan selalu menemui bilangan-bilangan prima, meskipun kemunculannya akan semakin jarang seiring dengan peningkatan jumlahnya yang kita periksa [16].

2. Metode Penelitian

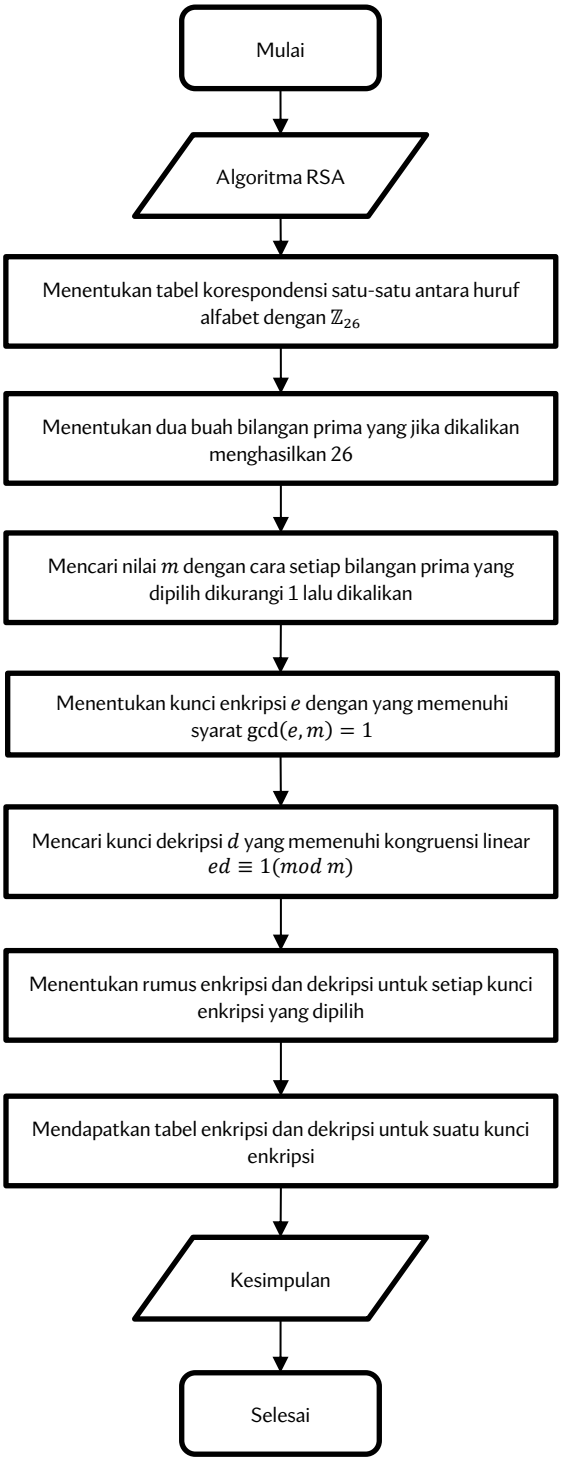
Penelitian yang disajikan pada artikel ini didasari dari sebuah pertanyaan yaitu bagaimanakah implementasi algoritma RSA pada kriptografi klasik (Gambar 1).

3. Hasil dan Diskusi

Algoritma RSA

1. Pilih a dan b bilangan prima sembarang. Jaga kerahasiaan a dan b ini.
2. Hitung $n = a \times b$. Besaran n boleh diketahui umum.
3. Hitung $m = (a-1) \times (b-1)$. Setelah menghiutng nilai m, a dan b dapat dihapus untuk menjaga kerahasiaan.
4. Pilih $e \in \mathbb{Z} \ni \gcd(e, m) = 1$.
5. Tentukan nilai d sebagai kunci dekripsi dengan menggunakan kongruensi $ed \equiv 1(mod\ m)$. Enkripsi pesan dilakukan dengan persamaan $E(p) = (p)^e\ mod\ n$, di mana p adalah blok plainteks, c adalah chiperteks yang dihasilkan, dan e adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa $0 \leq p < n$ untuk memastikan hasil perhitungan tetap dalam batas yang benar.
6. Proses dekripsi dilakukan dengan persamaan $D(c) = (c)^d\ mod\ n$.

Dengan menggunakan contoh sederhana yaitu 26 huruf alfabet, berikut tabel korespondensi satu-satu antara huruf alfabet dengan $\mathbb{Z}_{26}$ (Tabel 1).



Gambar 1. Diagram alur penelitian

Tabel 1. Korespondensi satu-satu antara alfabet dengan $\mathbb{Z}_{26}$

Huruf	A	B	C	D	E	F	G	H	I	J	K	L	M
$\mathbb{Z}_{26}$	0	1	2	3	4	5	6	7	8	9	10	11	12

Huruf	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$\mathbb{Z}_{26}$	13	14	15	16	17	18	19	20	21	22	23	24	25

Contoh 1.

1. Karena 26 merupakan perkalian dari dua buah bilangan prima yaitu 2 dan 13, maka pilih $a = 2$ dan $b = 13 \ni n = a \times b = 2 \times 13 = 26$
2. $m = (a-1) \times (b-1) = 1 \times 12 = 12$
3. Pilih $e = 5 \in \mathbb{Z}$ karena $\gcd(5,12) = 1$.
4. $ed \equiv 1(mod\ m)$, atau $5d \equiv 1(mod\ 12)$ sehingga $d \equiv 5(mod\ 12)$
5. $E(p) = p^5\ mod\ 26$
6. $D(c) = c^5\ mod\ 26$

Akan ditentukan tabel enkripsi dengan cara menentukan nilai $E(0),E(1),\ldots,E(25)$ (Tabel 2). Dengan demikian didapat tabel enkripsi seperti pada Tabel 3. Kemudian akan ditentukan tabel dekripsi dengan cara menentukan nilai $D(0),D(1),\ldots,D(25)$ (Tabel 4).

Tabel 2. Perhitungan persamaan enkripsi untuk kunci $e = 5$

"A"= 0, maka $E(0) = (0)^5 mod\ 26 = 0$ ="A"	"B"= 1, maka $E(1) = (1)^5 mod\ 26 = 1$ ="B"
"C"= 2, maka $E(2) = (2)^5 mod\ 26 = 6$ ="G"	"D"= 3, maka $E(3) = (3)^5 mod\ 26 = 9$ ="J"
"E"= 4, maka $E(4) = (4)^5 mod\ 26 = 10$ ="K"	"F"= 5, maka $E(5) = (5)^5 mod\ 26 = 5$ ="F"
"G"= 6, maka $E(6) = (6)^5 mod\ 26 = 2$ ="C"	"H"= 7, maka $E(7) = (7)^5 mod\ 26 = 11$ ="L"
"I"= 8, maka $E(8) = (8)^5 mod\ 26 = 8$ ="I"	"J"= 9, maka $E(9) = (9)^5 mod\ 26 = 3$ ="D"
"K"= 10, maka $E(10) = (10)^5 mod\ 26 = 4$ ="E"	"L"= 11, maka $E(11) = (11)^5 mod\ 26 = 7$ ="H"
"M"= 12, maka $E(12) = (12)^5 mod\ 26 = 12$ ="M"	"N"= 13, maka $E(13) = (13)^5 mod\ 26 = 13$ ="N"
"O"= 14, maka $E(14) = (14)^5 mod\ 26 = 14$ ="O"	"P"= 15, maka $E(15) = (15)^5 mod\ 26 = 19$ ="T"
"Q"= 16, maka $E(16) = (16)^5 mod\ 26 = 22$ ="W"	"R"= 17, maka $E(17) = (17)^5 mod\ 26 = 23$ ="X"
"S"= 18, maka $E(18) = (18)^5 mod\ 26 = 18$ ="S"	"T"= 19, maka $E(19) = (19)^5 mod\ 26 = 15$ ="P"
"U"= 20, maka $E(20) = (20)^5 mod\ 26 = 24$ ="Y"	"V"= 21, maka $E(21) = (21)^5 mod\ 26 = 21$ ="V"
"W"= 22, maka $E(22) = (22)^5 mod\ 26 = 16$ ="Q"	"X"= 23, maka $E(23) = (23)^5 mod\ 26 = 17$ ="R"
"Y"= 24, maka $E(24) = (24)^5 mod\ 26 = 20$ ="U"	"Z"= 25, maka $E(25) = (25)^5 mod\ 26 = 25$ ="Z"

Tabel 3. Tabel enkripsi algoritma RSA untuk kunci $e = 5$

$\mathcal{P}$	A	B	C	D	E	F	G	H	I	J	K	L	M
$\mathcal{C}$	A	B	G	J	K	F	C	L	I	D	E	H	M

$\mathcal{P}$	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$\mathcal{C}$	N	O	T	W	X	S	P	Y	V	Q	R	U	Z

Tabel 4. Perhitungan persamaan dekripsi untuk kunci $d = 5$

"A"= 0, maka $E(0) = (0)^5 mod\ 26 = 0$ ="A"	"B"= 1, maka $E(1) = (1)^5 mod\ 26 = 1$ ="B"
"C"= 2, maka $E(2) = (2)^5 mod\ 26 = 6$ ="G"	"D"= 3, maka $E(3) = (3)^5 mod\ 26 = 9$ ="J"
"E"= 4, maka $E(4) = (4)^5 mod\ 26 = 10$ ="K"	"F"= 5, maka $E(5) = (5)^5 mod\ 26 = 5$ ="F"
"G"= 6, maka $E(6) = (6)^5 mod\ 26 = 2$ ="C"	"H"= 7, maka $E(7) = (7)^5 mod\ 26 = 11$ ="L"
"I"= 8, maka $E(8) = (8)^5 mod\ 26 = 8$ ="I"	"J"= 9, maka $E(9) = (9)^5 mod\ 26 = 3$ ="D"
"K"= 10, maka $E(10) = (10)^5 mod\ 26 = 4$ ="E"	"L"= 11, maka $E(11) = (11)^5 mod\ 26 = 7$ ="H"
"M"= 12, maka $E(12) = (12)^5 mod\ 26 = 12$ ="M"	"N"= 13, maka $E(13) = (13)^5 mod\ 26 = 13$ ="N"

"O"= 14, maka $E(14) = (14)^5 mod\ 26 = 14$ ="O"	"P"= 15, maka $E(15) = (15)^5 mod\ 26 = 19$ ="T"
"Q"= 16, maka $E(16) = (16)^5 mod\ 26 = 22$ ="W"	"R"= 17, maka $E(17) = (17)^5 mod\ 26 = 23$ ="X"
"S"= 18, maka $E(18) = (18)^5 mod\ 26 = 18$ ="S"	"T"= 19, maka $E(19) = (19)^5 mod\ 26 = 15$ ="P"
"U"= 20, maka $E(20) = (20)^5 mod\ 26 = 24$ ="Y"	"V"= 21, maka $E(21) = (21)^5 mod\ 26 = 21$ ="V"
"W"= 22, maka $E(22) = (22)^5 mod\ 26 = 16$ ="Q"	"X"= 23, maka $E(23) = (23)^5 mod\ 26 = 17$ ="R"
"Y"= 24, maka $E(24) = (24)^5 mod\ 26 = 20$ ="U"	"Z"= 25, maka $E(25) = (25)^5 mod\ 26 = 25$ ="Z"

Tabel 5. Tabel dekripsi algoritma RSA untuk kunci $d = 5$

$\mathcal{C}$	A	B	C	D	E	F	G	H	I	J	K	L	M
$\mathcal{P}$	A	B	G	J	K	F	C	L	I	D	E	H	M

$\mathcal{C}$	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$\mathcal{P}$	N	O	T	W	X	S	P	Y	V	Q	R	U	Z

Dengan demikian didapat tabel dekripsi seperti pada Tabel 5.

Contoh 2.

1. Karena 26 merupakan perkalian dari dua buah bilangan prima yaitu 2 dan 13, maka pilih $a = 2$ dan $b = 13 \ni n = a \times b = 2 \times 13 = 26$
2. $m = (a-1) \times (b-1) = 1 \times 12 = 12$
3. Pilih $e = 7 \in \mathbb{Z}$ karena $\gcd(7,12) = 1$.
4. $ed \equiv 1(mod\ m)$, atau $7d \equiv 1(mod\ 12)$ sehingga $d \equiv 7(mod\ 12)$
5. $E(p) = p^7\ mod\ 26$
6. $D(c) = c^7\ mod\ 26$

Akan ditentukan tabel enkripsi dengan cara menentukan nilai $E(0)$,$E(1)$,...,$E(25)$ (Tabel 6). Dengan demikian didapat tabel enkripsi seperti pada Tabel 7.

Tabel 6. Perhitungan persamaan dekripsi untuk kunci $e = 5$

"A"= 0, maka $E(0) = (0)^7 mod\ 26 = 0$ ="A"	"B"= 1, maka $E(1) = (1)^7 mod\ 26 = 1$ ="B"
"C"= 2, maka $E(2) = (2)^7 mod\ 26 = 24$ ="Y"	"D"= 3, maka $E(3) = (3)^7 mod\ 26 = 3$ ="D"
"E"= 4, maka $E(4) = (4)^7 mod\ 26 = 4$ ="E"	"F"= 5, maka $E(5) = (5)^7 mod\ 26 = 21$ ="V"
"G"= 6, maka $E(6) = (6)^7 mod\ 26 = 20$ ="U"	"H"= 7, maka $E(7) = (7)^7 mod\ 26 = 19$ ="T"
"I"= 8, maka $E(8) = (8)^7 mod\ 26 = 18$ ="S"	"J"= 9, maka $E(9) = (9)^7 mod\ 26 = 9$ ="J"
"K"= 10, maka $E(10) = (10)^7 mod\ 26 = 10$ ="K"	"L"= 11, maka $E(11) = (11)^7 mod\ 26 = 15$ ="P"
"M"= 12, maka $E(12) = (12)^7 mod\ 26 = 12$ ="M"	"N"= 13, maka $E(13) = (13)^7 mod\ 26 = 13$ ="N"
"O"= 14, maka $E(14) = (14)^7 mod\ 26 = 14$ ="O"	"P"= 15, maka $E(15) = (15)^7 mod\ 26 = 11$ ="L"
"Q"= 16, maka $E(16) = (16)^7 mod\ 26 = 16$ ="Q"	"R"= 17, maka $E(17) = (17)^7 mod\ 26 = 17$ ="R"
"S"= 18, maka $E(18) = (18)^7 mod\ 26 = 8$ ="I"	"T"= 19, maka $E(19) = (19)^7 mod\ 26 = 7$ ="H"
"U"= 20, maka $E(20) = (20)^7 mod\ 26 = 6$ ="G"	"V"= 21, maka $E(21) = (21)^7 mod\ 26 = 5$ ="F"
"W"= 22, maka $E(22) = (22)^7 mod\ 26 = 22$ ="W"	"X"= 23, maka $E(23) = (23)^7 mod\ 26 = 23$ ="X"
"Y"= 24, maka $E(24) = (24)^7 mod\ 26 = 2$ ="C"	"Z"= 25, maka $E(25) = (25)^7 mod\ 26 = 25$ ="Z"

Akan ditentukan tabel dekripsi dengan cara menentukan nilai $D(0),D(1),\ldots,D(25)$ (Tabel 8). Dengan demikian didapat tabel dekripsi seperti pada Tabel 9.

4. Kesimpulan

Implementasi algoritma RSA dapat digunakan pada kriptografi klasik, namun dalam penerapannya memiliki banyak kekurangan pada proses enkripsi dan dekripsi karena banyak huruf yang berkorespondensi dengan dirinya sendiri. Perlu eksplorasi lebih lanjut terhadap implementasi algoritma RSA pada kriptografi klasik untuk mengatasi kekurangan pada proses enkripsi dan dekripsi dengan cara memperbanyak variasi huruf atau permutasi karakter sehingga bilangan bulat yang dipilih akan lebih besar, cara tersebut dapat meningkatkan keamanan dan efisiensi algoritma RSA pada kriptografi.

Tabel 7. Tabel enkripsi algoritma RSA untuk kunci $e = 7$

$\mathcal{P}$	A	B	C	D	E	F	G	H	I	J	K	L	M
$\mathcal{C}$	A	B	Y	D	E	V	U	T	S	J	K	P	M

$\mathcal{P}$	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$\mathcal{C}$	N	O	L	Q	R	I	H	G	F	W	X	C	Z

Tabel 8. Perhitungan persamaan dekripsi untuk kunci $d = 5$

"A"$= 0$, maka $D(0) = (0)^7 mod\ 26 = 0$ ="A"	"B"$= 1$, maka $D(1) = (1)^7 mod\ 26 = 1$ ="B"
"C"$= 2$, maka $D(2) = (2)^7 mod\ 26 = 24$ ="Y"	"D"$= 3$, maka $D(3) = (3)^7 mod\ 26 = 3$ ="D"
"E"$= 4$, maka $E(4) = (4)^7 mod\ 26 = 4$ ="E"	"F"$= 5$, maka $D(5) = (5)^7 mod\ 26 = 21$ ="V"
"G"$= 6$, maka $D(6) = (6)^7 mod\ 26 = 20$ ="U"	"H"$= 7$, maka $D(7) = (7)^7 mod\ 26 = 19$ ="T"
"I"$= 8$, maka $D(8) = (8)^7 mod\ 26 = 18$ ="S"	"J"$= 9$, maka $D(9) = (9)^7 mod\ 26 = 9$ ="J"
"K"$= 10$, maka $D(10) = (10)^7 mod\ 26 = 10$ ="K"	"L"$= 11$, maka $D(11) = (11)^7 mod\ 26 = 15$ ="P"
"M"$= 12$, maka $D(12) = (12)^7 mod\ 26 = 12$ ="M"	"N"$= 13$, maka $D(13) = (13)^7 mod\ 26 = 13$ ="N"
"O"$= 14$, maka $D(14) = (14)^7 mod\ 26 = 14$ ="O"	"P"$= 15$, maka $D(15) = (15)^7 mod\ 26 = 11$ ="L"
"Q"$= 16$, maka $D(16) = (16)^7 mod\ 26 = 16$ ="Q"	"R"$= 17$, maka $D(17) = (17)^7 mod\ 26 = 17$ ="R"
"S"$= 18$, maka $D(18) = (18)^7 mod\ 26 = 8$ ="I"	"T"$= 19$, maka $D(19) = (19)^7 mod\ 26 = 7$ ="H"
"U"$= 20$, maka $D(20) = (20)^7 mod\ 26 = 6$ ="G"	"V"$= 21$, maka $D(21) = (21)^7 mod\ 26 = 5$ ="F"
"W"$= 22$, maka $D(22) = (22)^7 mod\ 26 = 22$ ="W"	"X"$= 23$, maka $D(23) = (23)^7 mod\ 26 = 23$ ="X"
"Y"$= 24$, maka $D(24) = (24)^7 mod\ 26 = 2$ ="C"	"Z"$= 25$, maka $D(25) = (25)^7 mod\ 26 = 25$ ="Z"

Tabel 9. Tabel dekripsi algoritma RSA untuk kunci $d = 7$

$\mathcal{C}$	A	B	C	D	E	F	G	H	I	J	K	L	M
$\mathcal{P}$	A	B	Y	D	E	V	U	T	S	J	K	P	M

$\mathcal{C}$	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$\mathcal{P}$	N	O	L	Q	R	I	H	G	F	W	X	C	Z

Daftar Pustaka

1. Giri, B. E. 2022. Penerapan Kriptografi Dengan Metode Rsa Pada Internet Banking. *Timor Cerdas - Jurnal Teknologi Informasi, Manajemen Informasi dan Rekayasa Sistem Cerdas*, 1(1), 37–45.
2. Ginting, D. B. 2010. Peranan Aritmetika Modulo dan Bilangan Prima Pada Algoritma Kriptografi RSA(Rivest-Shamir-Adleman). *Jurnal Media Informatika*, *9*(2), 48-57.
3. Kukuh. 2006. *Penerapan Teori Bilangan Bulat dalam Kriptografi dan Aplikasinya dalam Kehidupan Sehari-hari*.
4. Suganda, A., Sinurat, S., & Ramadan, S. 2018. Penerapan Algoritma Sieve of Eratosthenes Untuk Pembangkit Bilangan Acak. *Pelita Informatika: Informasi dan Informatika,* 7(2), 145-148.
5. Gani, A. G. 2018. Pengamanan Komputer Menggunakan Kriptografi Cipher Block Chaining (CBC). *JSI (Jurnal sistem Informasi) Universitas Suryadarma,* 3(2), 79–100. https://doi.org/10.35968/JSI.V3I2.65.
6. Hans Delfs Helmut. 2007. *Introduction to Cryptography: Principles and Applications* (2nd ed.). New York: Springer.
7. Waruwu, T. S. 2015. *Analisa Cryptographically Secure Pseudorandom*.
8. Hasibuan, C. A. D. 2021. *Implementasi Kriptografi Dalam Penyisipan*.
9. Farid Fachrurozi, M. 2006. *Enkripsi Pesan Rahasia Menggunakan Algoritma (Advanced Encryption Standard) AES : RIJNDAEL*.
10. Rinaldi Munir. 2019. *Bahan Kuliah IF4020 Kriptografi: Kriptografi Klasik*.
11. Ginting, D. B. 2010. Peranan Aritmetika Modulo dan Bilangan Prima Pada Algoritma Kriptografi RSA(Rivest-Shamir-Adleman). *Jurnal Media Informatika*, *9*(2), 48-57.
12. Ronaldo Galman Kurniawan, I. 2010. *Bilangan Prima*.
13. Harahap, M. K. 2015. *Membangkitkan Bilangan Prima Mersenne di atas 512 Digit Menggunakan Kombinasi Eratosthenes dan Fermat Little Theorem Untuk Pendukung Kunci Publik RSA*.
14. Putri, A. S. 2015. *Teori Bilangan Prima Serta Uji Primalitas Teorema Lucas Dan Teorema Pocklington*.
15. Abidianto, H. 2009. *Bilangan Prima dan Aplikasinya dalam Bidang Informatika*. Retrieved from http://www.foxitsoftware.com
16. Ignatus Ronaldo Galman Kurniawan. 2010. *Bilangan Prima*.