PENGGUNAAN ALGORITMA RSA DENGAN METODE *THE SIEVE OF ERATOSTHENES* DALAM ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN

Disusun dan Diajukan sebagai Salah Satu Syarat Pengajuan Judul
Tugas Akhir/Skripsi pada Fakultas Sains Dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

DISUSUN OLEH :

NAMA	: YUSPRIDA SAKIMAN
NPM	: 1514370077
PROGRAM STUDI	: SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

YUSPRIDA SAKIMAN

PENGGUNAAN ALGORITMA RSA DENGAN METODE *THE SIEVE OF ERATOSTHENES* DALAM ENKRIPSI DAN DEKRIPSI PENGIRIMAN PESAN

2019

Seiring dengan berkembangnya teknologi pada masa ini mengubah cara masyarakat dalam melakukan pengiriman pesan, salah satu metode yang digunakan pada saat ini ialah saling mengirim pesan melalui media elektronik yang dapat dilakukan kapanpun dan dimanapun. Pada saat pengiriman pesan,terdapat beberapa teknik pengamanan agar pesan tetap aman ialah dengan menggunakan teknik kriptografi. Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Salah satu contoh kriptografi yang dapat diandalkan pada saat ini ialah RSA . Dimana keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima, maka dari itu dengan bantuan metode *The Sieve Of Eratosthenes* kita dapat menentukan bilangan prima acak yang digunakan untuk mengenkripsi pesan. Dengan ini maka akan dibuat sebuah aplikasi berbasis desktop yang akan dirancang dan dibuat dalam bahasa pemrograman *Visual Basic.NET 2010* sebagai sarana untuk meningkatkan keamanan pada saat pengiriman pesan.

Kata Kunci : Dekripsi, Desktop, Enkripsi, Kriptografi, RSA, *The Sieve Of Eratosthenes* ,*Visual Basic*.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena atas rahmat dan karunia-Nya yang telah diberikan kepada penulis sehingga dapat menyelesaikan skripsi ini yang merupakan salah satu syarat kelulusan di Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi. Adapun judul yang dibuat oleh penulis yaitu **"Penggunaan Algoritma RSA Dengan Metode *The Sieve Of Eratosthenes* Dalam Enkripsi Dan Dekripsi Pengiriman Pesan" .**

Dalam kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada banyak pihak yang telah membantu dalam penyelesaian penyusunan Skripsi ini. Penulis ingin mengucapkan terima kasih kepada :

1. Penghargaan yang sebesar-besarnya kepada kedua orang tua tercinta yang telah mendoakan serta memberikan dukungan moril maupun materil kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya.
2. Bapak Dr. H. Muhammad Isa Indrawan, SE, MM, selaku Rektor Universitas Pembangunanan Panca Budi Medan.
3. Ibu Sri Shindi Indira, ST.,M.Sc, selaku Dekan Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Eko Hariyanto,S.Kom.,M.Kom, selaku Ketua Program Studi Sistem Komputer Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan.
5. Dosen Pembimbing I Bapak Andysah Putra Utama Siahaan, S.Kom.,M.Kom. Ph.D.
6. Dosen Pembimbing II Bapak Dr.Muhammad Iqbal, S.Kom.,M.Kom.
7. Seluruh Dosen Dan Staf Pegawai Fakultas Ilmu Komputer yang telah banyak membantu dalam proses perkuliahan.
8. Teman-teman yang telah memberikan berbagai saran, inspirasi, dorongan, maupun doa sehingga penulis dapat menyelesaikan skripsi ini.

Penulis menyadari bahwa masih banyak kesalahan pada saat penulisan laporan skripsi ini, Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk penyempurnaan isi laporan skripsi ini.

Medan, 30 Oktober 2019
Penulis,

YUSPRIDA SAKIMAN
NPM : 1514370077

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya teknologi pada masa ini mengubah cara masyarakat dalam melakukan komunikasi, salah satu metode komunikasi yang digunakan pada saat ini ialah saling mengirim pesan melalui media elektronik yang dapat dilakukan kapanpun dan dimanapun. Pada saat berkomunikasi melalui pesan terdapat beberapa aspek yang harus diperhatikan, diantaranya keaslian data yang diterima, kerahasiaan dan otentikasi, oleh karena itu pada saat ini sangat dibutuhkan proses pengamankan pesan pada saat pengiriman dari orang yang tidak berhak mengetahui isi pesan tersebut, untuk menjaga kerahasiaan dan keaslian pesan tersebut, terdapat beberapa cara pengamanan pesan, salah satunya menggunakan teknik kriptografi.

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dengan menggunakan kunci dekripsi maka seseorang tersebut dapat mengembalikan data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa aturan tersendiri. (Kromodimoeljo, 2010:5).

Berdasarkan dari kunci yang digunakan kriptografi dibagi menjadi dua jenis yaitu algoritma simetris dan algoritma asimetris. Dimana algoritma simetris menggunakan

satu kunci untuk melakukan proses enkripsi dan dekripsinya, sedangkan algoritma asimetris memiliki dua kunci untuk melakukan proses enkripsi dan dekripsinya yaitu kunci umum *(public key)* yang bersifat tidak rahasia, kunci ini digunakan untuk mengenkripsi pesan dan kunci khusus *(private key)* yang bersifat rahasia digunakan untuk mendekripsi pesan. Salah satu contoh kriptografi algoritma asimetris yang dapat diandalkan pada saat ini yaitu RSA . Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima, maka dari itu dengan bantuan metode *The Sieve Of Eratosthenes* kita dapat menentukan bilangan prima acak yang digunakan untuk mengenkripsi pesan.

Pada penelitian ini penulis akan membuat buat sebuah aplikasi keamanan pesan berbasis desktop menggunakan algoritma RSA dan metode *the sieve of eratosthenes* untuk enkripsi dan dekripsi pesan. Adapun judul dari penelitian ini ialah **"Penggunaan Algoritma RSA dengan Metode *The Sieve Of Eratosthenes* dalam Enkripsi dan Dekripsi Pengiriman Pesan".**

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat disimpulkan beberapa rumusan masalah, diantaranya sebagai berikut:

1. Bagaimana mengimplementasikan algoritma RSA dengan bantuan metode *the sieve of eratosthenes* dalam enkripsi dan dekripsi pengiriman pesan?
2. Bagaimana proses pembentukan kunci publik (*public key)* dan kunci privat (*private key)* untuk enkripsi dan dekripsi pesan?

1.3 Batasan Masalah

Adapun batasan masalah dalam laporan skripsi ini ialah sebagai berikut:

1. Bahasa pemrograman yang digunakan dalam pembuatan aplikasi ini ialah *Visual basic.net 2010.*
2. Aplikasi yang dibuat nantinya akan berbasis *desktop*.
3. Pesan yang dapat diamankan hanya berupa teks.

1.4 Tujuan Penelitian

Adapun tujuan penelitian dalam laporan skripsi ini ialah sebagai berikut:

1. Untuk mengimplementasikan algoritma RSA dengan bantuan metode *the sieve of eratosthenes* dalam enkripsi dan dekripsi pengiriman pesan.
2. Untuk proses pembentukan kunci publik (*public key)* dan kunci privat (*private key)* untuk enkripsi dan dekripsi pesan.

1.5 Manfaat Penelitian

Berikut ini merupakan manfaat dari penelitian yang dilakukan ialah dengan bantuan aplikasi ini seseorang dapat mengirimkan pesan dengan lebih aman sehingga kerahasiaan pesan tetap terjaga.

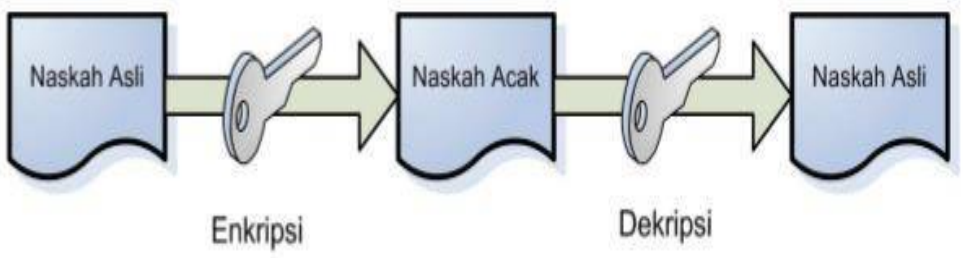
BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi merupakan ilmu tentang teknik pengamanan, dimana data akan diacak menggunakan kunci enkripsi menjadi data yang sulit dibaca apabila tidak memiliki kunci dekripsi. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan. (Kromodimoeljo, 2010:5).

Pada kriptografi, data asli disebut *plaintext* sedangkan data yang telah terenkripsi disebut *ciphertext*. Proses mengubah *plaintext* menjadi *cipertext* disebut enkripsi, sedangkan kebalikannya disebut dekripsi. Kedua proses enkripsi dan dekripsi membutuhkan penggunaan sejumlah informasi rahasia, yang sering disebut kunci *(key).*



Gambar 2.1: Proses enkripsi dan dekripsi
Sumber : Sentot Kromodimoeljo (2010)

Proses enkripsi adalah proses pengacakan naskah asli *(plaintext)* menjadi naskah acak *(ciphertext)* yang sulit untuk dibaca oleh seseorang yang tidak mempunyai kunci dekripsi. Satu cara untuk mendapatkan kembali naskah asli tentunya dengan menerka kunci dekripsi, jadi proses menerka kunci dekripsi harus menjadi sesuatu yang sulit. Tentunya naskah acak harus dapat didekripsi oleh seseorang yang mempunyai kunci dekripsi untuk mendapatkan kembali naskah asli. Pada awalnya kriptografi hanya digunakan untuk merahasiakan isi dari pesan teks, namun kini, seiring berkembangannya kriptografi dapat digunakan untuk mengamankan seluruh data yang bersifat digital.

Ada empat tujuan yang mendasar dari ilmu kriptograpi yang juga merupakan aspek keamanan informasi yaitu (Efrandi, et al, 2014):

1. Kerahasiaan

   Merupakan layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka atau menghapus informasi yang telah disandi. Integritas data adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah.

2. Integritas Data

   Sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak- pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubtitusian data lain kedalam data yang sebenarnya.

3. Autentikasi

Merupakan yang berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasiitu 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.

4. Non- repudiasi atau penyangkalan

Merupakan usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan atau yang membuat.

Dalam jurnal yang berjudul "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard*"* juga menjelaskan bahwa dalam kriptografi kita akan sering menemukan berbagai istilah atau *terminology* (Pabokory, et al,2015). Beberapa istilah yang harus diketahui yaitu:

1. Pesan

Pesan *(message)* adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintex)* atau teks jelas *cleartext*.

2. Pengirim dan Penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim *(sender)* adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima *(receiver)* adalah entitas yang menerima pesan.

3. Enkripsi dan dekripsi

Proses menyandikan *plainteks* menjadi *cipherteks* disebut enkripsi *(encryption)* atau *enciphering* . Sedangkan proses mengembalikan cipherteks menjadi plainteks semula disebut dekripsi *(decryption)* atau *deciphering* .

4. *Cipher* dan kunci

Algoritma kriptogarfi disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi *cipherteks*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut.

2.2 Algoritma Kriptografi

Algoritma merupakan langkah-langkah logis yang berurutan serta disusun secara sistematis untuk menyelesaikan masalah. Algoritma kriptografi merupakan langkah-langkah logis untuk menyembunyikan pesan dari orang-orang yang tidak berhak

mengetahui pesan tersebut (Ariyus, 2006:13). Berikut merupakan 3 fungsi dari algoritma kriptografi ialah sebagai berikut:

1. Enkripsi

   Enkripsi merupakan proses dari merubah pesan asli *(plaintext)* menjadi pesan acak atau pesan rahasia *(ciphertext).*

2. Dekripsi

   Dekripsi merupakan kebalikan dari enkripsi, yaitu proses pengembalian pesan acak *(ciphertext)* menjadi pesan asli *(plaintext).*

3. Kunci

   Kunci pada algoritma kriptografi berfungsi untuk mengenkripsi dan mendekripsi pesan, kunci yang terdapat pada algoritma asimetris terbagi menjadi 2 (dua) bagian yaitu kunci pribadi *(private key)* dan kunci umum *(public key).* Dalam kriptografi, hal yang harus dijaga ialah kunci, dikarenakan apabila kunci dari yang digunakan diketahui maka akan dengan mudah untuk mengetahui isi pesan.

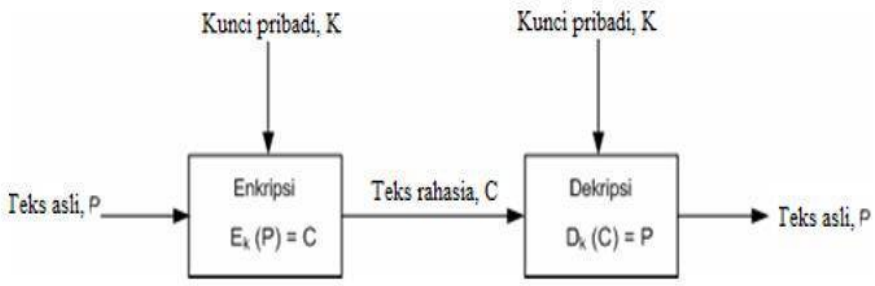
2.3 Macam-Macam Algoritma Kriptografi

Algoritma kriptografi dibagi menjadi 3 bagian yaitu algoritma simetris, algoritma asimetris dan *hash function.*

1. Algoritma Simetris

   Algoritma simetris dapat juga disebut dengan algoritma klasik, dikarena menggunakan kunci yang sama untuk proses enkripsi dan dekripsinya. Pada saat

melakukan pengiriman pesan menggunakan algoritma simetris, seseorang yang menerima pesan harus diberitahu kunci dari pesan tersebut agar sipenerima datap mendekripsikan pesan yang telah dikirim. Keamanan dari algoritma ini tergantung pada kerumitan kunci yang dipilih, semakin sulit kunci ditebak maka akan semakin sulit untuk mendekripsikan pesan.



Gambar 2.3.1: Proses algoritma simetris
Sumber : Jurnal Teknologi dan Sistem Komputer (2015)

a. Kelebihan algoritma kriptografi kunci simetris :
    1) Keamanan lebih baik saat pendistribusian kunci.
    2) Mudah dalam pengaturan kunci, karna kunci lebih sedikit.

b. Kelemahan algoritma kriptografi kunci simetris:
    1) Kecepatan yang lebih rendah bila dibandingkan dengan algoritma simetris.
    2) Lebih mudah ditebak, dikarenakan memiliki kunci yang sama untuk proses enkripsi dan dekripsi.

2. Algoritma Asimetris

Algoritma simetris dapat juga disebut dengan algoritma kunci *public*. Algoritma ini membutuhkan 2 kunci yang berbeda untuk melakukan enkripsi dan dekripsinya. Berikut merupakan 2 jenis kunci yang digunakan pada algoritma asimetris yaitu:

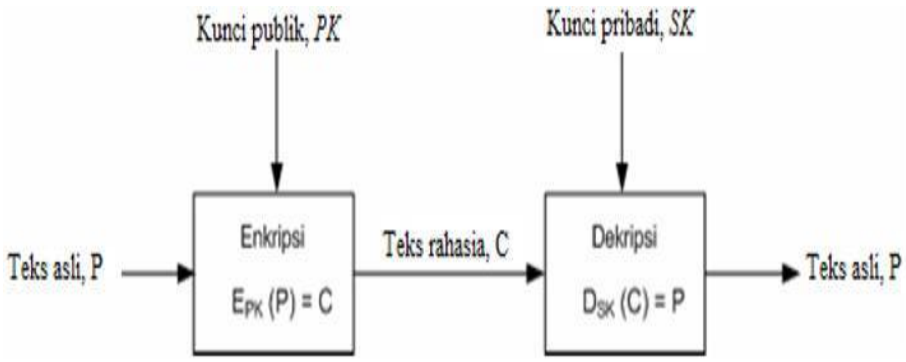
a. Kunci Umum *(Public key)*

Kunci ini bersifat tidak rahasia, artinya siapapun boleh mengetahuinya. Pada algoritma RSA kunci *public* digunakan untuk mengenkripsi pesan.

b. Kunci Pribadi *(Private key)*

Kunci ini bersifat rahasia, artinya hanya boleh diketahui oleh sipengirim pesan dan si penerima pesan. Pada algoritma RSA kunci pribadi digunakan untuk mendekripsi pesan.

Dengan menggunakan kunci umum seseorang dapat mengenkripsi pesan tetapi tidak dapat mendekripsikannya, hanya orang yang memiliki kunci pribadi yang dapat mendekripsi pesan tersebut. Algoritma asimetris bisa melakukan pengiriman pesan lebih aman dari pada algoritma simetris.



Gambar 2.3.2: Proses algoritma asimetris
Sumber : Jurnal Teknologi dan Sistem Komputer (2015)

3. *Hash Fuction* (Fungsi Hash)

Fungsi hash bisaanya digunakan untuk membuat sidik jari pada sebuah pesan. Sidik jari pada pesan merupakan suatu tanda yang menandakan bahwa pesan tersebut benar-benar dari orang yang dimaksud.

2.4 Algoritma RSA

Algoritma RSA termasuk salah satu algoritma kriptografi kunci publik (asimetris). Ditemukan pertama kali pada tahun 1977 oleh *Ron Rivest, Adi Shamir*, dan *Len Adleman*. Sebagai algoritma kunci publik, RSA memiliki dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmatika modulo. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan non prima menjadi faktor prima, semakin besar bilangan prima yang digunakan maka akan semakin aman. Berikut merupakan besaran yang digunakan pada algoritma RSA:

1. *p* dan *q* harus berupa bilangan prima yang dapat diambil secara acak (rahasia).
2. $n = p.q$ (tidak rahasia)
3. $\varphi(n) = (p-1)(q-1)$ (rahasia)
4. e = enkripsi (tidak rahasia)
5. d = dekripsi (rahasia)
6. PT = plaintext (rahasia)
7. CT = cipertext (tidak rahasia)

Selain dari sulitnya memfaktorkan bilangan prima, keamanan RSA juga berdasarkan pada fakta bahwa mengetahui n dan d secara umum tidak membantu untuk mencari e yaitu *inverse modulo* φ(n) dari d. Hal ini karena mengetahui n tidak membantu mencari φ(n) jika n tidak bisa diuraikan menjadi n = pq. Untuk menjaga keamanan tersebut, ada beberapa hal yang perlu diperhatikan dalam memilih p dan q:

1. Nilai p harus cukup jauh dari nilai q.
2. Sebaiknya panjang dari p harus berbeda beberapa digit dari q. Jika nilai p terlalu dekat dengan nilai q, maka Fermat factorization dapat digunakan untuk menguraikan n = pq.
3. Sebaiknya gcd (p−1,q−1) tidak terlalu besar.
4. Sebaiknya p−1 dan q−1 mempunyai faktor prima yang besar

2.5 Metode *The Sieve Of Eratosthenes*

Metode *the sieve of Eratosthenes* berfungsi untuk mengeliminasi sejumlah bilangan yang telah ditentukan untuk mencari bilangan prima. Cara kerja metode ini adalah dengan melakukan eliminasi terhadap bilangan yang bukan bilangan prima untuk menyaring suatu kumpulan bilangan prima. Berikut ini adalah langkah-langkah dari metode *the sieve of Eratosthenes :*

1. Tulis daftar bilangan – bilangan yang akan diuji primalitasnya, dari 2 hingga bilangan terbesar yang ditentukan. Sebut saja daftar nilai ini sebagai daftar A.
2. Tandai angka dua dari daftar A dan pindahkan ke dalam daftar yang lain sebagai menampung bilangan –bilangan pembangkit bilangan acak yang akan cari.
3. Coret semua angka yang merupakan kelipatan dari 2 dalam daftar A.
4. Angka yang ditemukan berikutnya dalam daftar A adalah bilangan pembangkit bilangan acak. Tandai dan pindahkan ke daftar B
5. Coret semua angka yang merupakan kelipatan bilangan pembangkit bilangan acak dalam daftar A tadi, angka yang telah dicoret sebelumnya tidak perlu dicoret lagi. Ulangi proses sempai semua nilai selesai. Berikut perhitungan yang dimaksud yakni dengan algoritma Sieve Of Eratoshneses:

Tabel 2.5 : Daftar bilangan A

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Tabel 2.5.1 : Daftar bilangan B

2	3	5	7	11
13	17	19	23	29
31	37	41	43	47
53	59	61	67	71
73	79	83	89	97

Daftar bilangan A merupakan daftar bilangan yang akan dieliminasi untuk menemukan bilangan prima sedangkan daftar bilangan B merupakan daftar bilangan prima yang ditemukan.

a. Warna merah merupakan kelipatan 2.
b. Warna kuning merupakan kelipatan 3.
c. Warna hijau merupakan kelipatan 5.
d. Warna biru merupakan kelipatan 7.
e. Warna ungu merupakan kelipatan 13.

2.6 Kode ASCII *(American Standard Code for Information Interchange)*

Kode Standar Amerika untuk pertukaran informasi atau ASCII *(American Standard Code for Information Interchange)* merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Kode ini selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambakan satu angka 0 sebagai *bit significant* paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas.

Dalam pengkodean kode ASCII memanfaatkan 8 bit. Pada saat ini kode ASCII telah tergantikan oleh kode UNICODE *(Universal Code).* UNICODE dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan

kodekode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya. Berikut merupakan tabel dari kode ASCII diantaranya:

Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	00	000	0000000	NUL (null character)	32	20	040	0100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	01	001	0000001	SOH (start of header)	33	21	041	0100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	02	002	0000010	STX (start of text)	34	22	042	0100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	03	003	0000011	ETX (end of text)	35	23	043	0100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	04	004	0000100	EOT (end of transmission)	36	24	044	0100100	$	68	44	104	1000100	D	100	64	144	1100100	d
5	05	005	0000101	ENQ (enquiry)	37	25	045	0100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	06	006	0000110	ACK (acknowledge)	38	26	046	0100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	07	007	0000111	BEL (bell (ring))	39	27	047	0100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	08	010	0001000	BS (backspace)	40	28	050	0101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	09	011	0001001	HT (horizontal tab)	41	29	051	0101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	012	0001010	LF (line feed)	42	2A	052	0101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	013	0001011	VT (vertical tab)	43	2B	053	0101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	014	0001100	FF (form feed)	44	2C	054	0101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	015	0001101	CR (carriage return)	45	2D	055	0101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	016	0001110	SO (shift out)	46	2E	056	0101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	017	0001111	SI (shift in)	47	2F	057	0101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	020	0010000	DLE (data link escape)	48	30	060	0110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	021	0010001	DC1 (device control 1)	49	31	061	0110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	022	0010010	DC2 (device control 2)	50	32	062	0110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	023	0010011	DC3 (device control 3)	51	33	063	0110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	024	0010100	DC4 (device control 4)	52	34	064	0110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	025	0010101	NAK (negative acknowledge)	53	35	065	0110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	026	0010110	SYN (synchronize)	54	36	066	0110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	027	0010111	ETB (end transmission block)	55	37	067	0110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	030	0011000	CAN (cancel)	56	38	070	0111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	031	0011001	EM (end of medium)	57	39	071	0111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	032	0011010	SUB (substitute)	58	3A	072	0111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	033	0011011	ESC (escape)	59	3B	073	0111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	034	0011100	FS (file separator)	60	3C	074	0111100	<	92	5C	134	1011100	\	124	7C	174	1111100	|
29	1D	035	0011101	GS (group separator)	61	3D	075	0111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	036	0011110	RS (record separator)	62	3E	076	0111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	037	0011111	US (unit separator)	63	3F	077	0111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL

Gambar 2.6 Tabel ASCII 127

2.7 *Unified Modeling Language* (UML)

UML (*Unified Modeling Language*) merupakan salah satu bahasa standar yang banyak digunakan pada dunia industri untuk membuat analisis, desain dan

menggambarkan arsitektur dalam program berorientasi objek (OOP). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan dan membangun sistem perangkat lunak (Sukamto, 2016:133).

Didalam UML (*Unified Modeling Language*) terdapat beberapa pembagian kategori yaitu sebagai berikut:

1. *Structure diagrams* merupakan kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang akan dibuat.
2. *Behavior diagrams* merupakan kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* merupakan kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain pada sebuah sistem.

2.8 Diagram-Diagram UML

Berikut ini beberapa diagram *Unified Modeling Language* (UML) yang biasa digunakan untuk pembuatan sistem ialah:

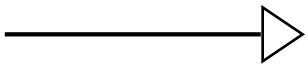
1. Diagram kelas *(Class Diagram)*

   Diagram ini menggambarkan struktur sistem dari segi mendefinisikan kelas-kelas yang akan dibuat untuk memodelkan sebuah sistem. Kelas memiliki bagian-bagian yang disebut atribut dan metode/operasi. Atribut ialah variabel yang dimiliki suatu kelas, sedangkan metode/operasi merupakan fungsi yang

dimiliki suatu kelas. Berikut merupakan simbol-simbol yang terdapat pada diagram kelas diantaranya:

Tabel 2.8.1 Simbol-simbol diagram kelas

No.	Simbol	Keterangan
1	Kelas Nama_kelas +atribut +operasi()	Kelas pada struktur sistem
2	Antarmuka/*interface*  nama_interface	-
3	Asosiasi/ *association* 	Relasi antarkelas dengan makna umum. Asosiasi biasanya disertai *multiplicity*.
4	Asosiasi berarah / *directed association* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.

5	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesifikasi (umum khusus).
6	Kebergantungan/*dependency* 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7	Agregasi/*aggregation* 	Relasi antarkelas dengan makna semua-bagian *(whole-part).*

Sumber : Rosa Arianti Sukamto (2016)

2. Diagram Objek *(Object Diagram)*

Diagram objek merupakan diagram yang digunakan untuk menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek pada suatu sistem. Diagram ini juga berfungsi untuk mendefinisikan contoh nilai atau isi dari atribut tiap kelas. Berikut merupakan simbol-simbol dari diagram objek yaitu:

Tabel 2.8.2 Simbol-simbol diagram objek

No.	Simbol	Keterangan
1.	Objek name_object : nama_kelas Atribut = nilai	Objek dari kelas yang berjalan saat sistem dijalankan.
2.	Link	Relasi antar objek

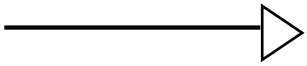
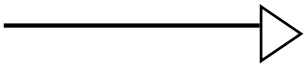
Sumber : Rosa Arianti Sukamto (2016)

3. *Use case* Diagram

Diagram *use case* merupakan diagram yang digunakan untuk menggambarkan suatu interaksi antara satu atau lebih aktor pada suatu sistem. Terdapat 2 bagian utama pada diagram *use case* yaitu aktor dan *use case*, dimana aktor merupakan orang, proses atau sistem yang berinteraksi didalam sebuah sistem, sedangkan *use case* merupakan unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut merupakan simbol-simbol dari diagram *use case* yaitu:

Tabel 2.8.3 Simbol-simbol diagram *use case*

No.	Simbol	Keterangan
1	*Use case* Nama *use case*	unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan kata kerja diawal.

2	Aktor/*actor* 	orang, proses atau sistem yang berinteraksi didalam sebuah sistem
3	Asosiasi/*association* 	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* memiliki interaksi dengan aktor.
4	Ekstensi/*extend* 	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.
5	Generalisasi/*generalization* 	Hubungan generalisasi dan spesifikasi (umum-khusus) antara 2 buah *use case* dimana fungsi yang satu lebih umum dari fungsi lainnya.
6	Menggunakan/*include/uses* <<include>> <<uses>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.

Sumber : Rosa Arianti Sukamto (2016)

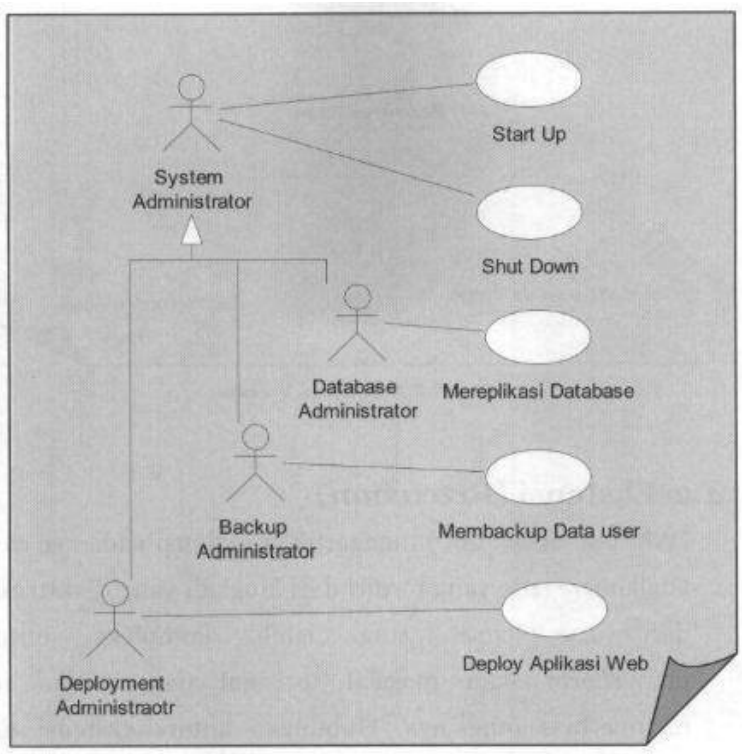
Berikut merupakan komponen pembentuk diagram *use case* ialah sebagai berikut:

a. Aktor : Menggambarkan pihak-pihak yang berperan dalam sistem.

b. *Use Case* : Aktivitas yang dijalankan oleh sistem.

c. Hubungan *(Link)* : Aktor mana saja yang terlibat dalam diagram *use case.*

Gambar dibawah ini merupakan salah satu contoh bentuk dari diagram *use case.*

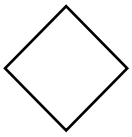
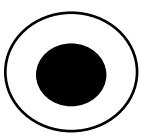


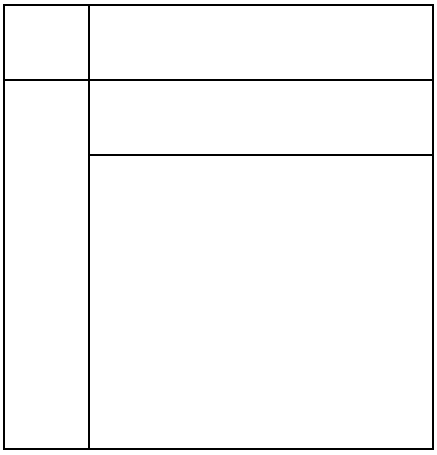
Gambar 2.8.1 Simbol-simbol diagram *use case*
Sumber : Rosa Arianti Sukamto (2016)

4. Diagram Aktivitas *(Activity Diagram)*

Diagram aktivitas ini digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari suatu sistem pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

Tabel 2.8.4 Simbol-simbol diagram aktivitas

No.	Simbol	Keterangan
1	Status awal 	Status awal aktivitas sistem.
2	Aktivitas 	Aktivitas yang dilakukan sistem.
3	Percabangan/*dicision* 	Digunakan apabila aktivitas lebih dari satu.
4	Penggabungan/*join* 	Digunakan apabila lebih dari satu aktivitas digabungkan
5	Status akhir 	Status akhir aktivitas sistem.

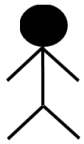
6	*Swimlane* 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.

Sumber : Rosa Arianti Sukamto (2016)

5. *Sequence Diagrams*

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada diagram sekuen.

Tabel 2.8.5 Simbol-simbol diagram *sequence*

No.	Simbol	Keterangan
1.	Aktor 	Aktor yang menggambarkan pihak-pihak yang berperan dalam sistem.
2.	Garis hidup/*lifeline*	Menyatakan kehidupan suatu objek yang digunakan sistem.
3.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
4.	Pesan 	Digunakan untuk menyatakan suatu objek.

Sumber : Rosa Arianti Sukamto (2016)

2.9 *Microsoft Visual Studio* 2010

Microsoft Visual Studio 2010 merupakan sebuah software yang digunakan untuk pembangunan maupun pengembangan aplikasi, diantaranya aplikasi berbasis *desktop*, web serta windows mobile. *Visual studio* ini dapat digunakan untuk membuat aplikasi yang berbasis *desktop* yang merupakan *platform windows*, namun juga dapat berjalan

dalam bentuk *Microsoft Intermediate Language* diatas *.Net Framework*. Selain itu *Visual Studio* juga dapat digunakan untuk membuat aplikasi yang dapat dijalankan diatas *windows mobile* yang berjalan diatas *.Net Compact Framework* (Yesputra, 2017:1).

Visual Bacic 2010 ialah pengembangan terintegrasi atau IDE yang dikembangkan menggunakan bahasa pemrograman BASIC. Sintaks yang sederhana dan fleksibel membuat bahasa pemrograman ini menjadi lebih mudah digunakan.

2.10 *Flowchart*

Flowchart merupakan gambaran berbentuk suatu grafik yang disertai lagkah-langkah dan urutan suatu prosedur dari suatu program. *Flowchart* dapat membantu proses analisis, perancangan dan pengkodean untuk memecahkan masalah kedalam bagian-bagian yang lebih kecil untuk mengoperasiannya.

Berikut merupakan beberapa jenis-jenis *Flowchart* antara lain sebagai berikut:

1. *Flowchart* Sistem

   Merupakan bagan yang menggambarkan alur kerja suatu sistem secara keseluruhan. *Flowchart* sistem terdiri dari data yang mengalir pada sistem dan terdapat proses transformasi data.

2. *Flowchart* Program

   Flowchart ini biasanya dihasilkan dari *Flowchart* sistem. *Flowchart* program merupakan bagan yang menerangkan tentang kegiatan yang dilakukan secara rinci dan dengan urutan yang tepat.

3. *Flowchart* Dokumen

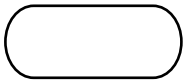
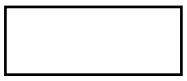
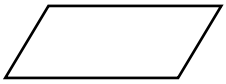
Flowchart ini merupakan *Flowchart* yang menggambarkan alur dari data yang ditulis disistem. Fungsi *Flowchart* ini ialah untuk menelusuri alur sistem dari satu bagian ke bagian lainnya.

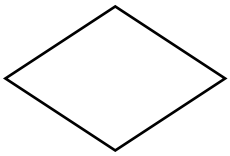
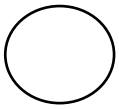
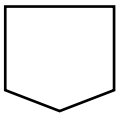
4. *Flowchart* Proses

Flowchart proses merupakan suatu teknik dekripsi rekayasa yang memecahkan masalah dengan langkah-langkah sesuai pada suatu sistem. Pada analisa sistem, *Flowchart* proses sangat efektif untuk digunakan menelusuri alur suatu laporan atau form.

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart* :

Tabel. 2.10 Simbol *Flowchart*

Simbol	**Nama**	**Arti**
	*Terminator*	Permulaan atau akhir program
	Garis Alir	Arah aliran program
	*Preparation*	Proses perhitungan dan pengolahan data.
	*Input/Output*	Masukan/keluaran

	*Decision*	Simbol untuk menyatakan perbandingan dan penyeleksian data
	*Predefined Program*	Permulaan sub program/proses menjalankan sub program.
	*On Page Connector*	Penghubung pada bagian flowchat yang terdapat dihalaman yang sama.
	*Off Page Connector*	Penghubung pada bagian flowchat yang terdapat dihalaman yang berbeda.

2.11 Perbandingan Penelitian Terdahulu

Berikut merupakan perbandingan terhadap penelitian-penelitian terdahulu yang bersangkutan dengan judul yang diambil.

Tabel. 2.11 Perbandingan Penelitian Terdahulu

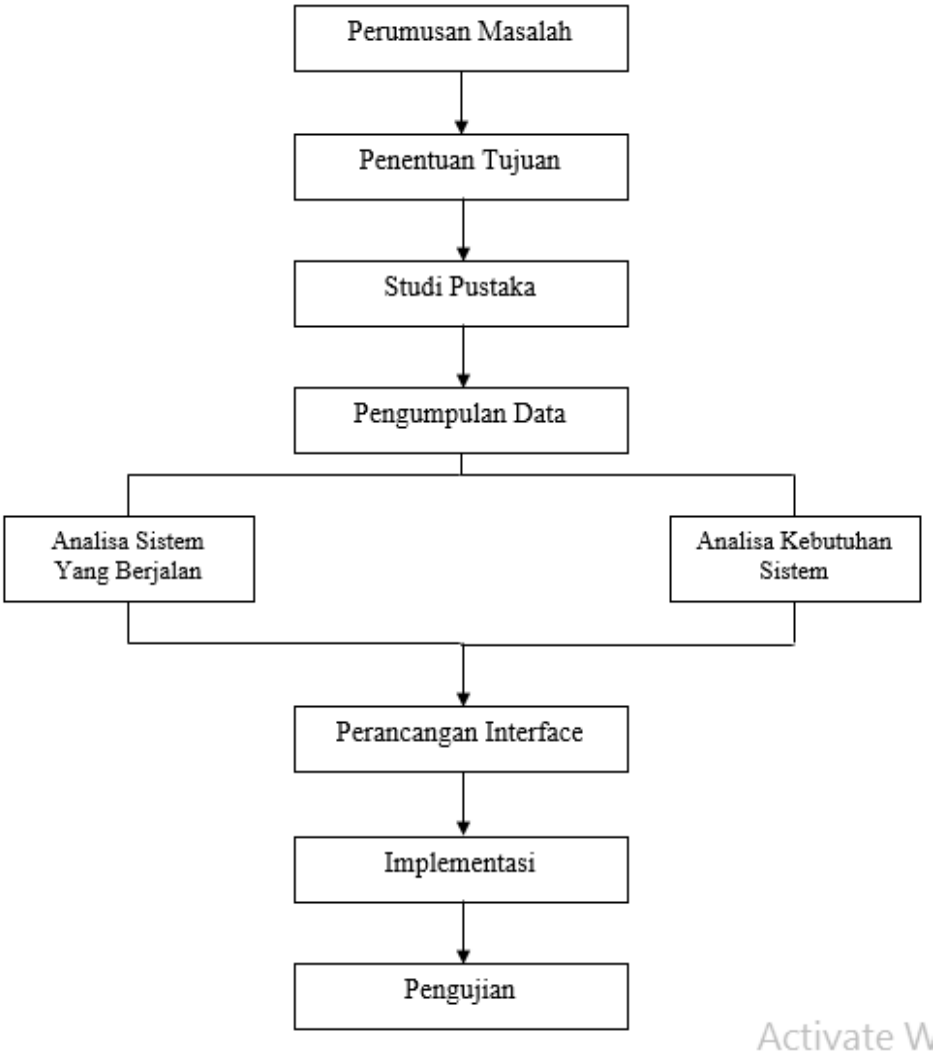
No.	Nama Penulis	Judul	Tahun	Hasil
1	Albert Ginting, R. Rizal Isnanto, Ike P. Windasari.	Implementasi Algoritma Kriptografi RSA Untuk Enkripsi Dan Dekripsi Email.	2015	Membuat aplikasi keamanan isi pesan email menggunakan Algoritma RSA, aplikasi hanya akan mengamankan isi pesan email bukan mengamankan jalur transfer email.
2	M. Safri Lubis, M. Andri Budiman, Karina Lolo Manik.	Penggunaan Algoritma RSA Dengan Metode The Sieve Of Eratosthenes Dalam Enkripsi Dan Dekripsi Pengiriman Email.	2013	Membuat aplikasi keamanan email menggunakan algoritma RSA serta metode The Sieve Of Eratosthenes.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Penerapan algoritma RSA dengan metode The Sieve Of Eratosthenes dalam enkripsi dan dekripsi pesan adalah sebagai berikut:



Gambar 3.1 : Tahapan penelitian
Sumber : Berdasarkan Rancangan Penulis (2019)

1. Analisis Sistem

   Pada tahapan ini akan dilakukan analisis terhadap sistem yang berjalan dan analisis terhadap kebutuhan sistem.

2. Perancangan Interface

   Pada tahapan ini akan dilakukan perancangan interface dari sistem yang akan dibangun, ada pun rancangannya terdiri dari rancangan *Flowchart* sistem dan rancangan dari halaman utama dan halaman proses.

3. Implementasi

   Tahapan implementasi akan dilakukan pada saat sistem sudah selesai dirancang, kemudian siap untuk dijadikan sebuah aplikasi.

4. Pengujian

   Pada tahapan ini akan dilakukan pengujian terhadap aplikasi yang telah dibangun, apakah aplikasi dapat berjalan dengan lancer dan telah dapat menyelesaikan masalah yang menjadi fokus.

3.2 Metode Pengumpulan Data

Pengumpulan data adalah pencarian terhadap susuatu kerena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 3, yaitu :

1. Pengamatan (*Observation*)

   Penulis melakukan pengamatan terhadap beberapa aplikasi pengiriman pesan teks untuk mengamati proses keamanan yang sudah dibuat sebelumnya.

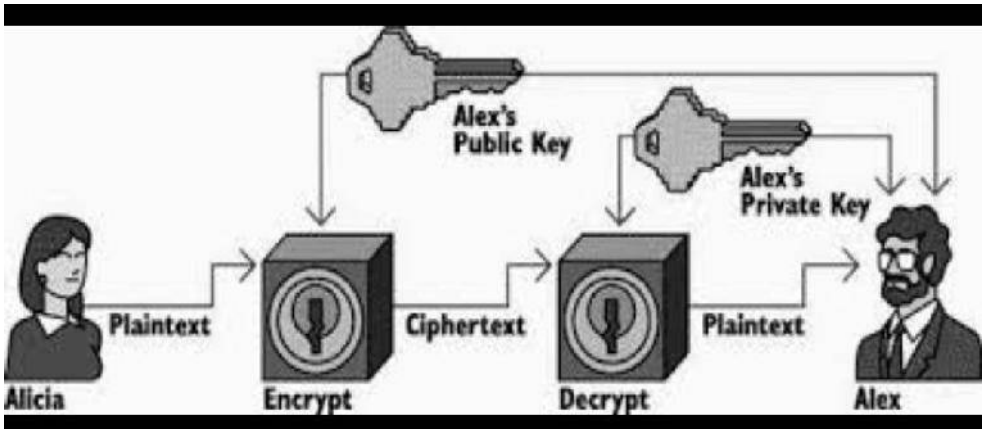
2. Penelitian Kepustakaan (*Library Research*)

Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus, maupun perpustakaan umum, juga melakukan pencarian lewat internet, dengan mengunjungi situs-situs seperti *google Book online* yang dapat membantu pembahasan materi.

3.3 Analisis Sistem Sedang Berjalan

Algoritma RSA merupakan salah satu algoritma asimetris dimana dalam proses enkripsi dan dekripsinya memiliki dua buah kunci, yaitu kunci umum (*Public Key*) dan Kunci khusus (*Private key*). Untuk melakukan sebuah pengiriman pesan menggunakan algoritma RSA dapat dimulai dengan penerima pesan membangkitkan kunci public dan kunci privat kemudian kunci public dikirimkan kepada si pengirim pesan untuk melakukan proses enkripsi pesan, setelah pesan terenkripsi kemudian dikirim kembali kepada sipenerima pesan untuk didekripsikan menggunakan kunci privat agar pesan yang disampaikan dapat dimengerti oleh sipenerima.

Pemberitahuan kata kunci dari penerma ke pengirim menggunakan media yang umum digunakan oleh banyak orang. Proses pengirimannya dapat dilihat pada gambar dibawah ini.



Gambar 3.3 : Skema Cara Kerja RSA

Skema RSA sendiri mengadopsi dari skema block cipher, dimana sebelum dilakukan enkripsi, plainteks yang ada dibagi – bagi menjadi blok – blok dengan panjang yang sama, dimana plainteks dan cipherteksnya berupa integer (bilangan bulat) antara 1 hingga n, dimana n berukuran biasanya sebesar 1024 bit, dan panjang bloknya sendiri berukuran lebih kecil atau sama dengan log(n) +1 dengan basis 2. Fungsi enkripsi dan dekripsinya dijabarkan dalam fungsi berikut :

P = bilangan prima untuk membangkitkan kunci

Q = bilangan prima untuk membangkitkan kunci

N = Modulo pembanding

T = Totient

E = Kunci enkripsi

D = kunci dekripsi

PT = Plaintext

CT = Ciphertext

Pi = Isi Pesan

Visual basic 2010 akan menjadi sarana untuk menciptakan perangkat lunak ini. Pada analisa proses ini penggunaan digunakan sebagai metode yang didalamnya terdapat kombinasi dari algoritma *RSA*. Algoritma *RSA* digunakan oleh pengirim untuk mengenkripsi pesan yang akan dikirimkan.

Langkah-langkah mengenkripsi data menggunakan algoritma RSA yaitu, Kedua pihak harus mengetahui nilai e dan nilai n ini, dan salah satu pihak harus memilki d untuk melakukan dekripsi terhadap hasil enkripsi dengan menggunakan public key e.

Sebelum memulai penggunaan RSA ini, terlebih dahulu kita harus memiliki bahan-bahan dasar sebagai berikut :

1. p, q = 2 bilangan prima yang dirahasiakan
2. n, dari hasil p*q
3. Φ(n), dengan (p-1)*(q-1)
4. e, dengan ketentuan gcd (Φ(n);e) =1
5. d, dengan (Φ(n)*K+1)/e

Proses dari algoritma RSA sendiri terdiri dari 3 proses, sebagai berikut :

1. **Proses Pembangkitan Kunci :**

   P = **13**

   Q = **31**

   N = p * q

   = 13 * 31

   = **403**

T = (p – 1)*(q – 1)

= (13 – 1)*(31 - 1)

= 12 * 30

= **360**

E = gcd (Φ(n);e) =1 , e = **7**

= gcd (7, 120) = 1

D = (Φ(n)*K+1)/e

Untuk mendapatkan nilai d dengan cara mencobakan k = 1, 2, 3, …, hingga memperoleh d yang bulat misal K =2

D = (360*2+1)/7

= (721)/7

= **103**

Hasil 103 merupakan bilangan bulat sehingga boleh digunakan sebagai nilai dari d.

2. Proses Enkripsi

Setelah didapat perhitungan di atas, maka akan dilakukan enkripsi plaintext PT = MARIBELAJAR. Pertama-tama plaintext tersebut diubah menjadi format ASCII sebagai berikut:

Karakter	**M**	**A**	**R**	**I**	**B**	**E**	**L**	**A**	**J**	**A**	**R**
ASCII (PT)	**77**	**65**	**82**	**73**	**66**	**69**	**76**	**65**	**74**	**65**	**82**

Setelah dibagi perblock, maka akan dihitung menggunakan rumus:

CT = PT ^ e mod n

C1 = 77 ^ 7 mod 403

= 16048523266853 mod 403

= 116

C2 = 65 ^ 7 mod 403

= 4902227890625 mod 403

= 234

C3 = 82 ^ 7 mod 403

= 24928547056768 mod 403

= 173

C4 = 73 ^ 7 mod 403

= 11047398519097 mod 403

= 44

C5 = 66 ^ 7 mod 403

= 545516071056 mod 403

= 326

C6 = 69 ^ 7 mod 403

= 7446353252589 mod 403

= 121

C7 = 76 ^ 7 mod 403

= 14645194571776 mod 403

= 236

C8 = 65 ^ 7 mod 403

= 4902227890625 mod 403

= 234

C9 = 74 ^ 7 mod 403

= 12151280273024

= 334

C10 = 65 ^ 7 mod 403

= 4902227890625 mod 403

= 234

C11 = 82 ^ 7 mod 403

= 24928547056768 mod 403

= 173

CT	**116**	**234**	**173**	**44**	**326**	**121**	**236**	**234**	**334**	**234**	**173**

3. Proses Dekripsi

Setelah chipertext dari kata MARIBELAJAR didapat, untuk mengubahnya kembali jadi plaintext menggunakan dekripsi dengan rumus PT = CT ^ d mod n.

P1 = 116 ^ 103 mod 403

= 435685283209927739935161393322612070499559451149296100194438872412309922769110245007086228272151954525456975066793351441140153629697601307127848922945271755161108178113025111033177017639848939331880120275903184896 mod 403

= 77

P2 = 234 ^ 103 mod 403

= 106962935502985798424129296215049498116063945817395
654195850151838155500341645000106337294008637759934
468232576120988888063744071640870460989831508205180
211165778257678867191807971330578269359819171513653
477723400111265345250680565783824737239O4 mod 403

= 65

P3 = 173 ^ 103 mod 403

= 330178372325271928602440497374745826101840526643
424856510247697985398668691472392850720174769804
375545889463814674477975441404081272918140724794
053287617710358827246431252888873496258027975063
546295761502985099242295240039975579717 mod 403

= 82

P4 = 44 ^ 103 mod 403

= 188636477517838740970936221867613076416713958965l
012467002758103474511583590057905424776043326537l
183425069323507066622936418037086042241347409034O
30626680951825376477184 mod 403

= 73

P5 = 326 ^ 103 mod 403

= 726796466142135569541962521858394580054831313794
302472968398790670402065455845646313860914403882
351902328617831119290410472083289893852609049971
732239029200554229645556839373490640699218275636
721348344498170999808140068707092082799780151310
0228988734564990976 mod 403

= 66

P6 = 121 ^ 103 mod 403

= 336428781471572209763157438294301829844457754894413921050616171682663785720800901017738364135614355250957609067699469935916675564017157671756346924807034032627789173753930934047663199475445949783069048216782821835 61 mod 403

= 69

P7 = 236 ^ 103 mod 403

= 257001883350076147539443172159956141470630532147950304442731020539894678056415803693492299018988126927972032539891288914538640869437191829051890262110650326601584932618231797416066350065612168504181736536877924061827204712578879171444186084802 5

6 mod 403

= 76

P8 = 234 ^ 103 mod 403

= 106962935502985798424129296215049498116063945817395654195850151838155500341645000106337294008637759934468232576120988888063744071640870460989831508205180211165778257678867191807971330578269359819171513653477723400111265345250680565783824737239042473723904 mod 403

= 65

P9 = 334 ^ 103 mod 403

= 882848292906641497885839198635292702246941068672850046031528836205510857218259406525057006264845290576995872563761530031792859384894012565613937703338312072886025116340564628072681014108960433543876494462718692719915367886080307442195278425755406083491406807042473723904 mod 403

= 74

P10 = 234 ^ 103 mod 403

= 106962935502985798424129296215049498116063945817395654195850151838155500341645000106337294008637759934468232576120988888063744071640870460989831508205180211165778257678867191807971330578269359819171513653477723400111265345250680565783824737239042473723904 mod 403

739565419585015183815550034164500010633729400 86

377599344682325761209888880637440716408704609 89

831508205180211165778257678867191807971330578 26

935981917151365347772340011126534525068056578 38

2473723904 mod 403

= 65

P11 = 173 ^ 103 mod 403

= 330178372325271928602440497374745826101840526643

424856510247697985398668691472392850720174769804

375545889463814674477975441404081272918140724794

053287617710358827246431252888873496258027975063

546295761502985099242295240039975579717 mod 403

= 82

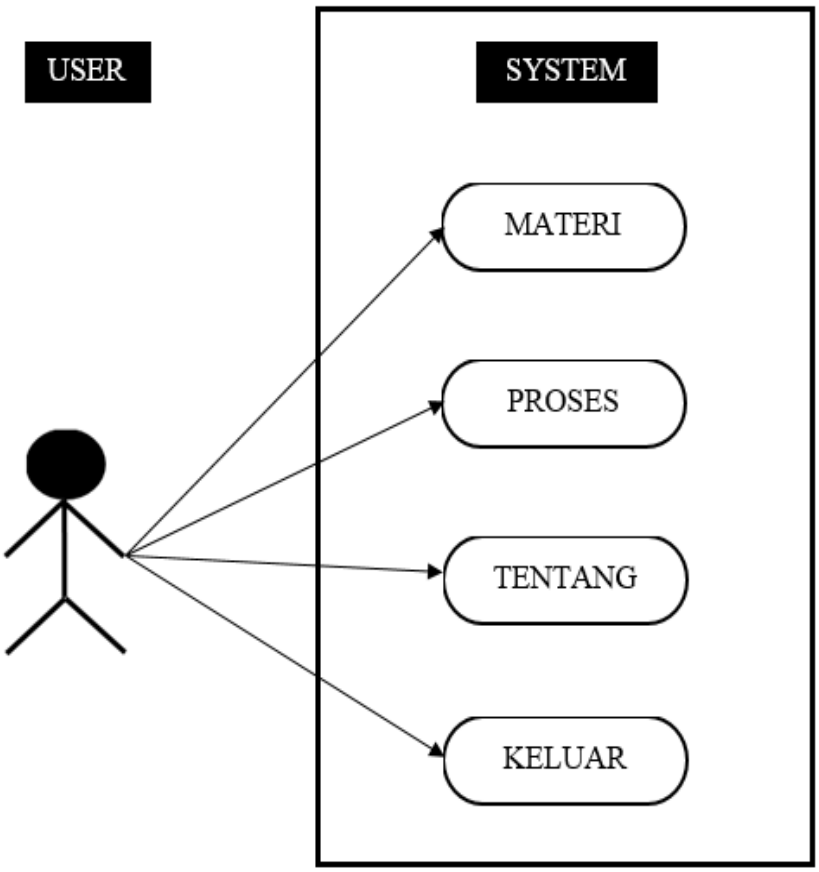
PT	**77**	**65**	**82**	**73**	**66**	**69**	**76**	**65**	**74**	**65**	**82**

3.4 Rancangan Penelitian

1. Rancangan UML

   a. *Use Case* Diagram

   Diagram *Use Case* berikut ini menggambarkan interaksi antara *user* dan sistem yaitu:



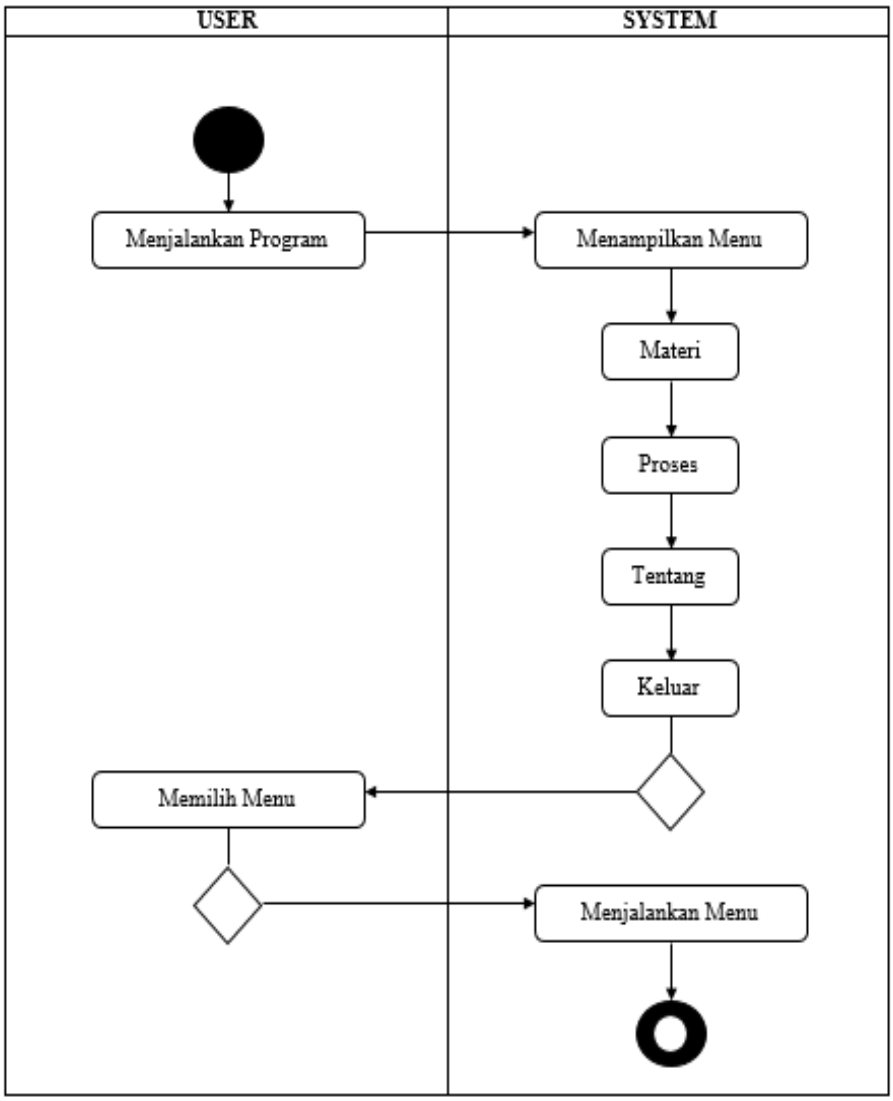
Gambar 3.4.1 *Use Case* Diagram
Sumber : Berdasarkan Rancangan Penulis (2019)

Keterangan :

Pada gambar diagram diatas menerangkan bahwa *user*/pengguna sebagai aktor sedangkan *system* sebagai media, proses, tentang dan keluar.

b. Diagram Aktivitas

Diagram aktivitas berikut ini menggambarkan kegiatan-kegiatan yang terjadi didalam sistem sampai sistem berhenti berjalan.



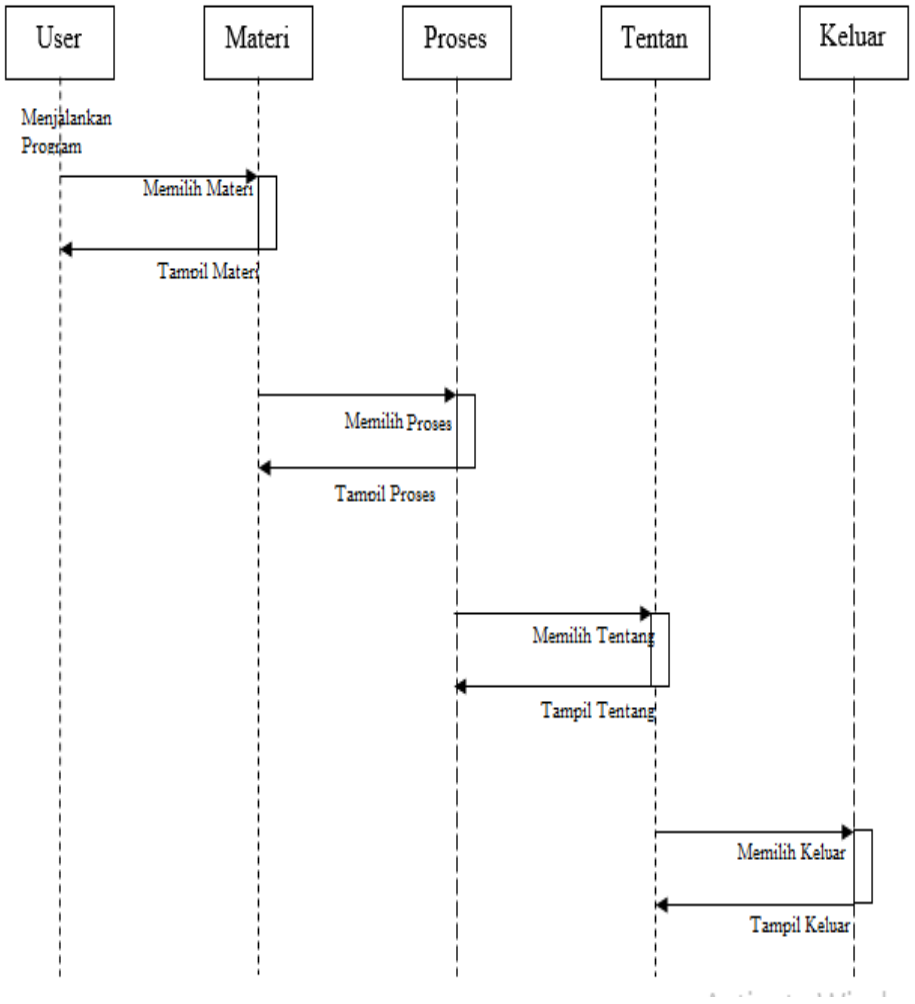
Gambar 3.4.2 *Activity* Diagram
Sumber : Berdasarkan Rancangan Penulis (2019)

Keterangan :

User memulai dengan menjalankan program, kemudian system menanggapi dengan menampilkan menu yang terdapat pilihan materi, proses, tentang dan keluar. Kemudian user melanjutkan dengan memilih menu yang telah ditampilkan, selanjutnya system menanggapi dengan menjalankan menu yang dipilih.

c. Sequence Diagram

Berikut merupakan gambaran dari sequence diagram:



Gambar 3.4.3 *Sequence* Diagram
Sumber : Berdasarkan Rancangan Penulis (2019)

Keterangan :

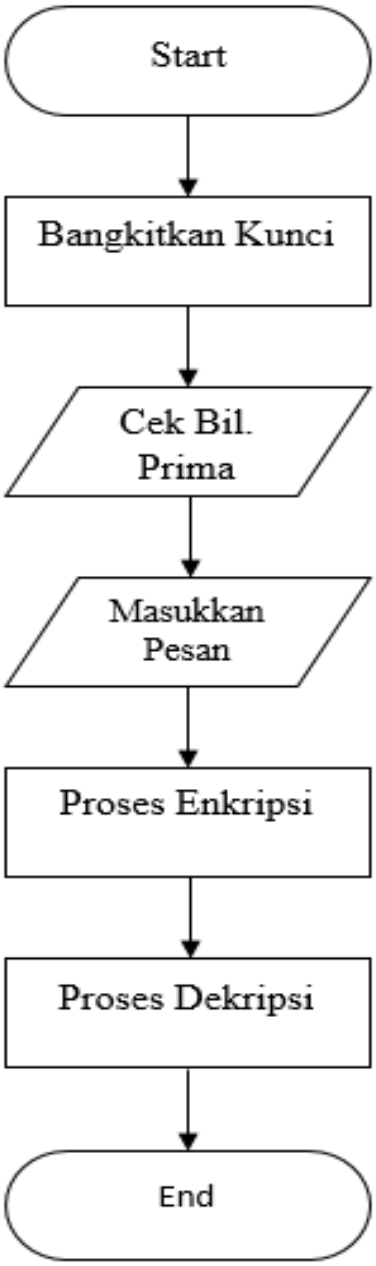
Pada diagram diatas menjelaskan bahwa saat user mulai menjalankan program dan memilih menu materi, maka sistem akan menampilkan menu materi. Saat user memilih proses maka system akan menampilkan

menu proses, begitu juga saat user memilih menu tentang dan keluar, maka system akan menampilkan menu tersebut.

2. Rancangan *Flowchart* Program

Flowchart merupakan langkah awal pembuatan program. Dengan adanya *Flowchart* urutan proses kegiatan menjadi lebih jelas. Bila terdapat penambahan proses maka dapat dilakukan lebih mudah. Setelah *Flowchart* selesai disusun, selanjutnya pemrogram (programmer) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

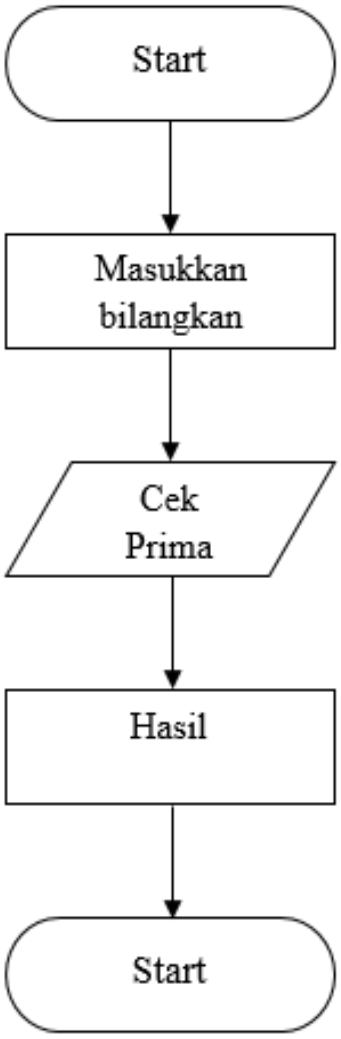
Flowchat program ini menggambarkan bagaimana cara program yang dibuat dapat berjalan, mulai dari membangkitkan kunci yang telah berisi bilangan prima yang akan ditampilkan secara acak, setelah melewati proses pengeliminasian menggunakan metode *The Sieve Of Eratosthenes*. Kemudian masukkan pesan yang akan dienkripsi setelah itu lanjutkan dengan proses enkripsi. Berikut merupakan bentuk dari flowchart program yaitu:



Gambar 3.4.4 *Flowchart* Sistem
Sumber : Berdasarkan Rancangan Penulis (2019)

Didalam program yang dibuat terdapat beberapa opsi tambahan seperti tombol untuk mengecek bilangan prima, button ini dapat digunakan untuk

menguji, apakah nilai yang muncul berupa bilangan prima atau tidak. Berikut merupakan rancangan flowchart dari cek prima.

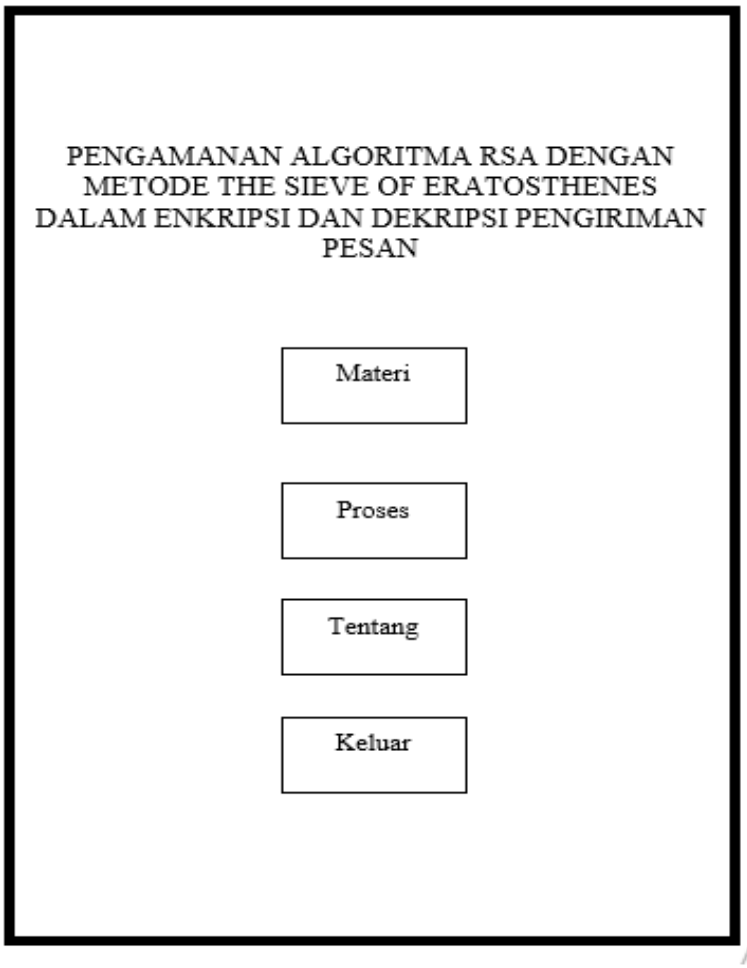


Gambar 3.4.5 *Flowchart* Uji Prima
Sumber : Berdasarkan Rancangan Penulis (2019)

3. Perancangan Antarmuka

   a. Rancangan Halaman Menu Utama

Form ini berisi tombol-tombol seperti menu Materi, Proses, Tentang, dan Keluar.



Gambar 3.4.6 Tampilan Halaman Utama
Sumber : Berdasarkan Rancangan Penulis (2019)

Pada tampilan di atas terdapat 4 tombol yaitu Materi, Proses, Tentang dan keluar.

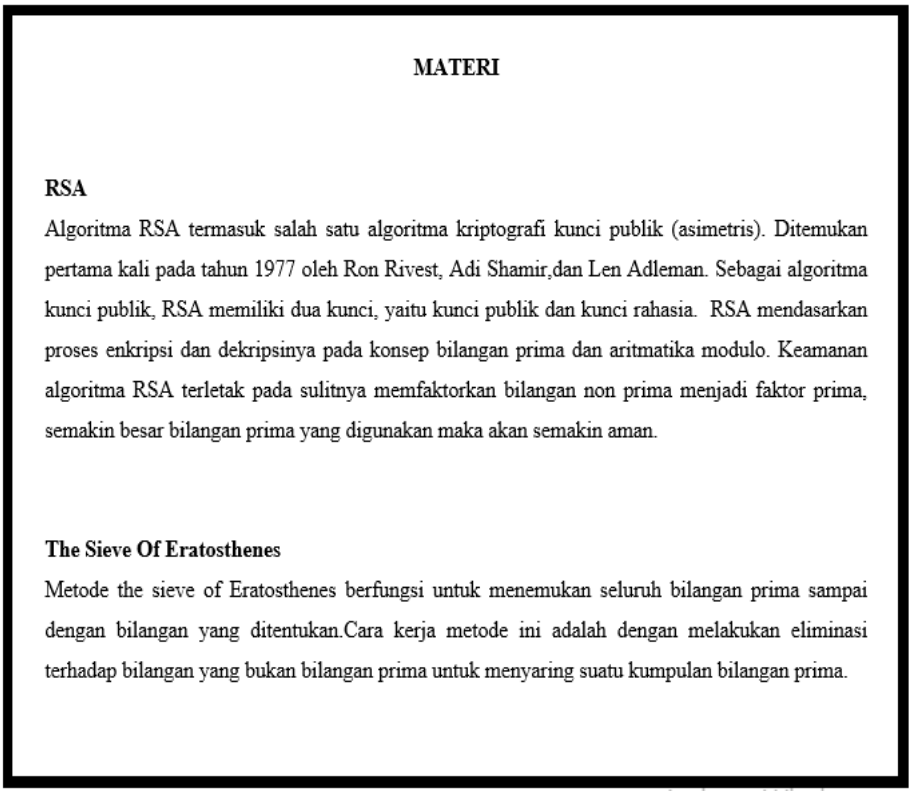
a. Tombol Materi berfungsi untuk menghubungkan pengguna ke form materi.
b. Tombol Proses berfungsi untuk menghubungkan pengguna ke form proses.

c. Tombol Tentang berfungsi untuk menghubungkan pengguna ke form tentang.

d. Tombol Keluar berfungsi untuk keluar dari program.

b. Rancangan Halaman Materi

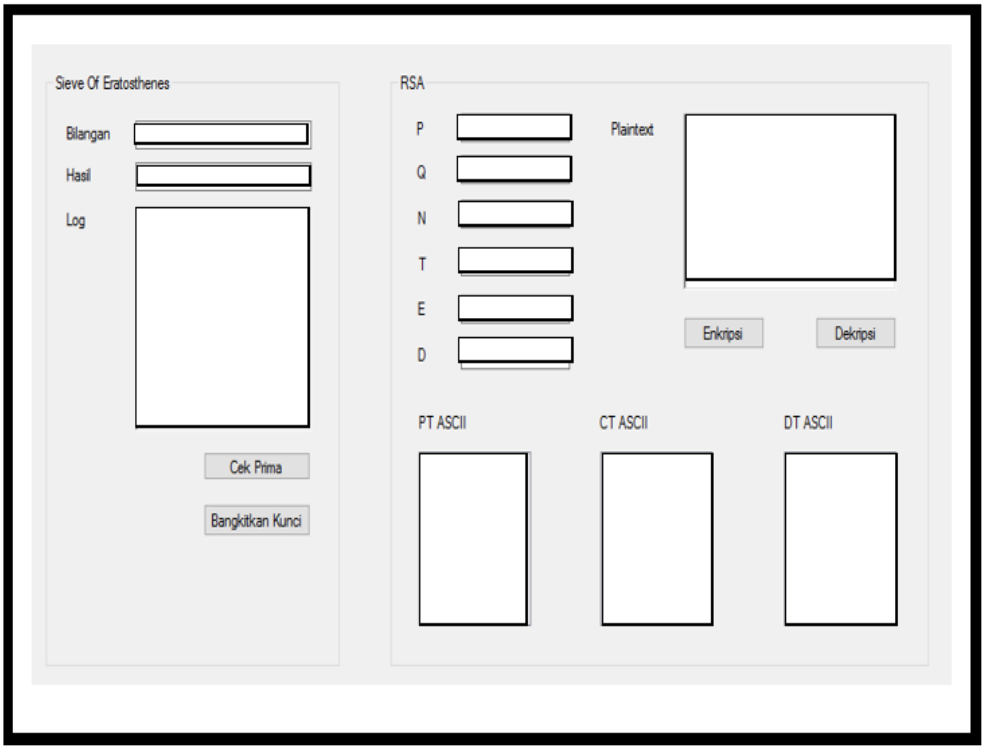
Form ini digunakan untuk menjelaskan mengenai algoritma RSA mulai dari jenis kunci yang digunakan, persyaratan menggunakan algoritma RSA serta kelebihan dari algoritma RSA. Selain mengenai algoritma RSA, pada form ini dijelaskan mengenai metode yang digunakan yaitu Metode *The Sieve Of Eratosthenes.*



Gambar 3.4.7 Tampilan Halaman Materi
Sumber : Berdasarkan Rancangan Penulis (2019)

c. Rancangan Halaman Proses

Berisi mengenai proses yang akan dikerjakan. Pengguna memasukkan teks yang akan dienkripsi atau *plaintext* ke dalam tombol masukan *plaintext* kemudian pilih tombol bangkitkan kunci, kunci yang dibangkitkan telah melalui proses eliminasi menggunakan metode *The Sieve Of Eratosthenes* serta kunci ini yang akan digunakan untuk proses enkripsi dan dekripsi pesan. Setelah itu, ditekan tombol Enkripsi untuk melakukan proses penyandian serta tombol Dekripsi untuk mengembalikan pesan yang telah disandi ke bentuk semula.



Gambar 3.4.7 Tampilan Halaman Materi
Sumber : Berdasarkan Rancangan Penulis (2019)

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Spesifikasi Minimum Hardware dan Software

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen – komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukkan yan diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukkan sehingga menghasilkan suatu keluaran yang diinginkan.

1. Analisis Perangkat Keras (Hardware)

   Perangkat keras minimum yang digunakan untuk membangun Sistem Informasi Penjualan ini adalah

   a. Processor berkecepatan 2.0 Ghz
   b. RAM 2 Gb
   c. Hardisk minimal 10 Gb untuk menyimpan data
   d. Keyboard dan Mouse
   e. Monitor 14

2. Analisis Perangkat Lunak (Software)

   Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai. Yaitu berupa perangkat lunak (software) yang

dirancang untuk memudahkan dalam pembangunan dan menjalankan sisten nantinya. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

a. Microsoft Windows 7 , Windows XP sebagai sistem operasi

b. Visual Studio 2010

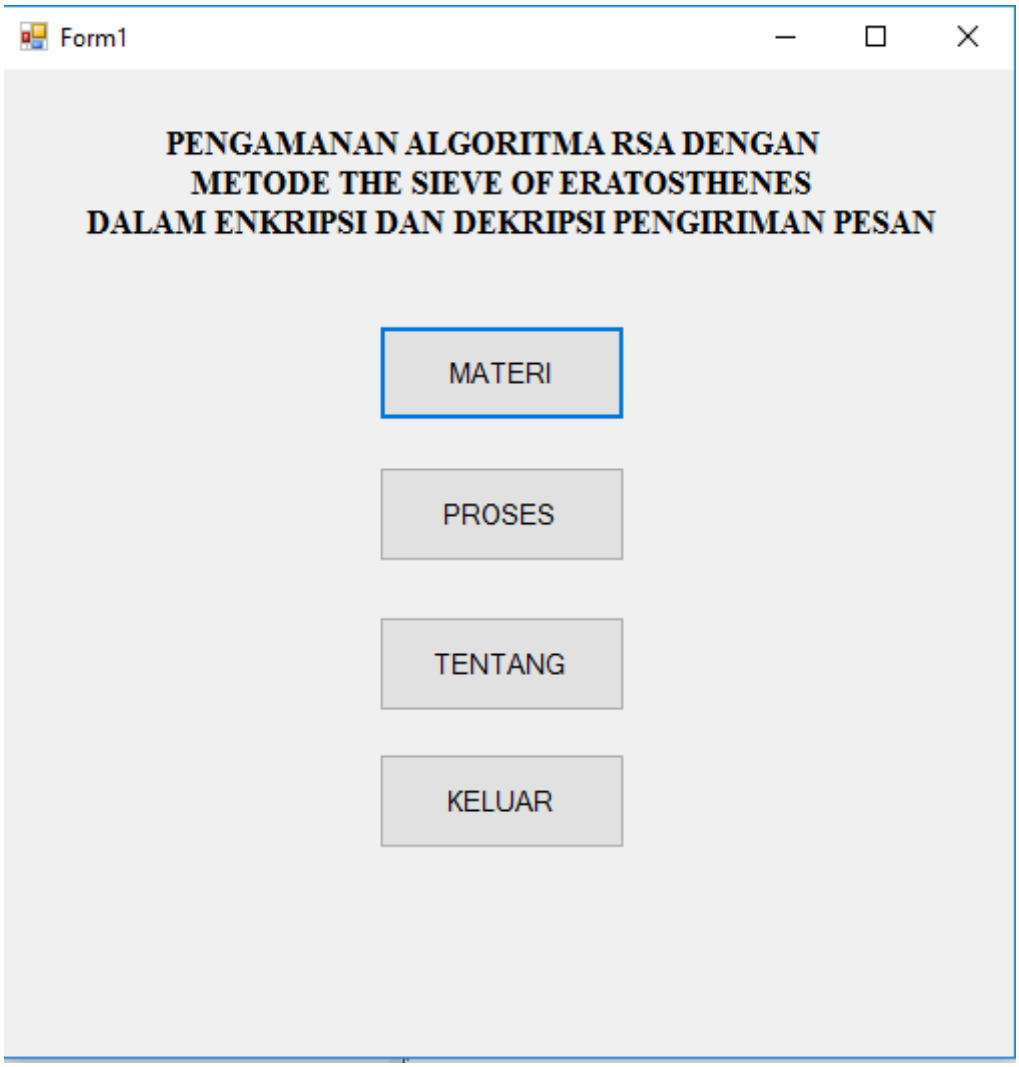
4.2 Pengujian Aplikasi Dan Pembahasan

Perlunya dilakukan pengujian sistem antara lain untuk memuji apakah sistem yang dibuat dapat berjalan lancer dan sesuai dengan yang diinginkan. Dari pengujian sistem ini akan terlihat bagian-bagian dari sistem yang mungkin masih bermasalah dan perlunya perbaikan ulang.

Pengujian dilakukan dengan memasukkan pesan teks dapat berupa angka dan huruf untuk selanjutnya diproses oleh aplikasi apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan pada pengujian aplikasi ini adalah simulasi pengenkripsian teks serta mendekripsikannya kembali dengan menggunakan metode Algoritma RSA dan bantuan metode *The Sieve Of Eratosthenes*.

1. Tampilan Awal/ Home

Tampilan pada gambar 4.2.1 merupakan tampilan awal ketika aplikasi dijalankan. Pada form ini pengguna dapat memilih untuk membuka beberapa form lainnya seperti tombol Materi yang akan mengarahkan pengguna menuju form yang menjelaskan tentang algoritma yang digunakan untuk membuat aplikasi ini serta beberapa tombol lainnya.



Gambar 4.2.1 Tampilan Awal/ Home
Sumber : Hasil Percobaan Aplikasi (2019)

Pada tampilan di atas terdapat 4 tombol yaitu Materi, Proses, Tentang dan keluar.

a. Tombol Materi berfungsi untuk menghubungkan pengguna ke form materi, yang berisi mengenai penjelasan tentang algoritma RSA beserta metode *The Sieve Of Eratosthenes*.

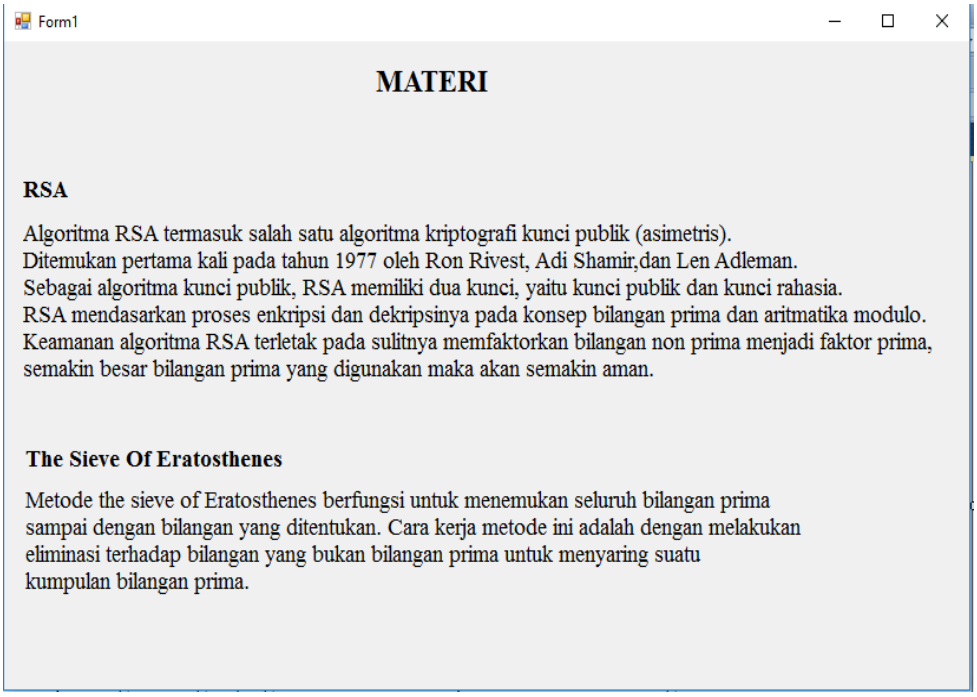
b. Tombol Proses berfungsi untuk menghubungkan pengguna ke form proses, yang berisi tentang proses pembangkitan kunci dan proses mengenkripsi dan mendekripsi pesan.

c. Tombol Tentang berfungsi untuk menghubungkan pengguna ke form tentang, yang berisi judul dan biodata penulis.

d. Tombol Keluar berfungsi untuk keluar dari program.

2. Tampilan Halaman Materi Aplikasi

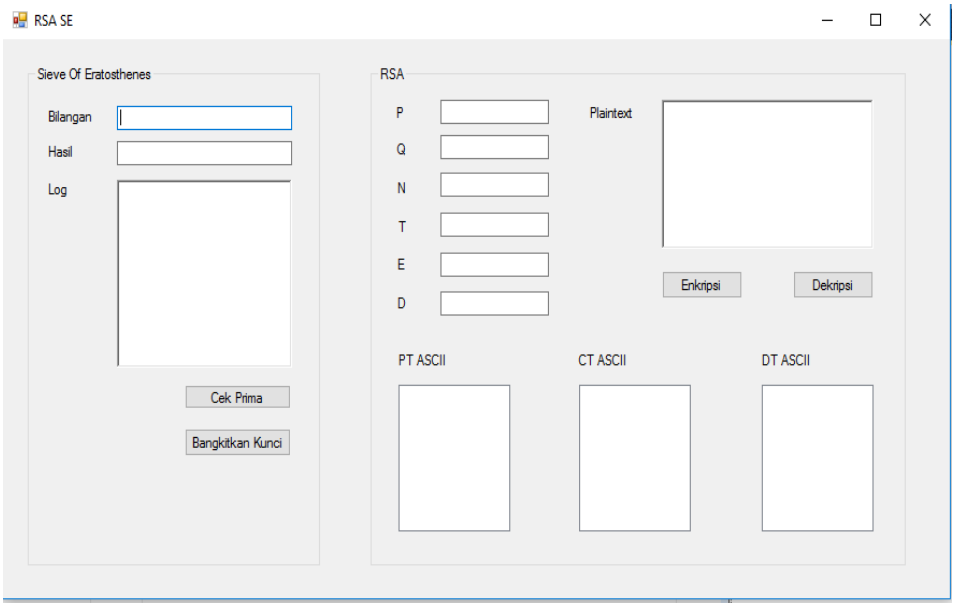
Tampilan materi aplikasi merupakan tampilan halaman atau form yang berisi tentang penjelasan mengenai algoritma yang digunakan pada aplikasi ini.



Gambar 4.2.2 Tampilan Halaman Materi Aplikasi
Sumber : Hasil Percobaan Aplikasi (2019)

3. Tampilan Halaman Proses

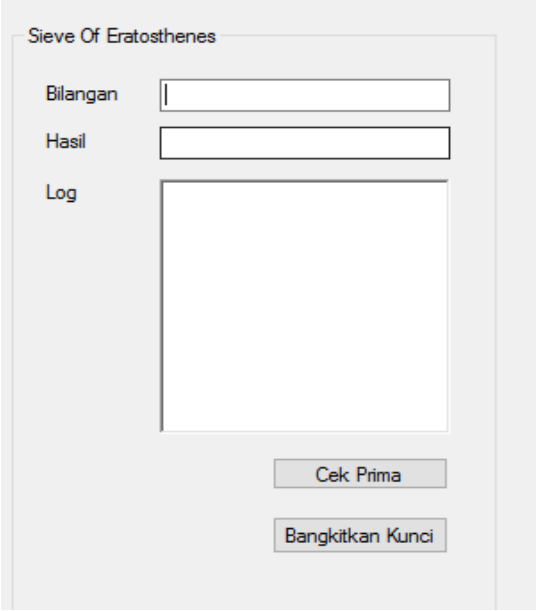
Tampilan berikut ini merupakan tampilan proses dari pembentukan kunci, penentuan nilai serta mengenkripsi dan mendekripsikan teks. Algoritma RSA merupakan protokol yang menjamin tidak adanya pertukaran kunci antara pihak-pihak yang melakukan enkripsi dan dekripsi. Kedua belah pihak menggunakan kunci mereka masing-masing untuk mengenkripsi pesan dan kemudian untuk mendekripsi pesan tanpa perlu mengetahui kunci yang lainnya.



Gambar 4.2.3 Tampilan Halaman Proses
Sumber : Hasil Percobaan Aplikasi (2019)

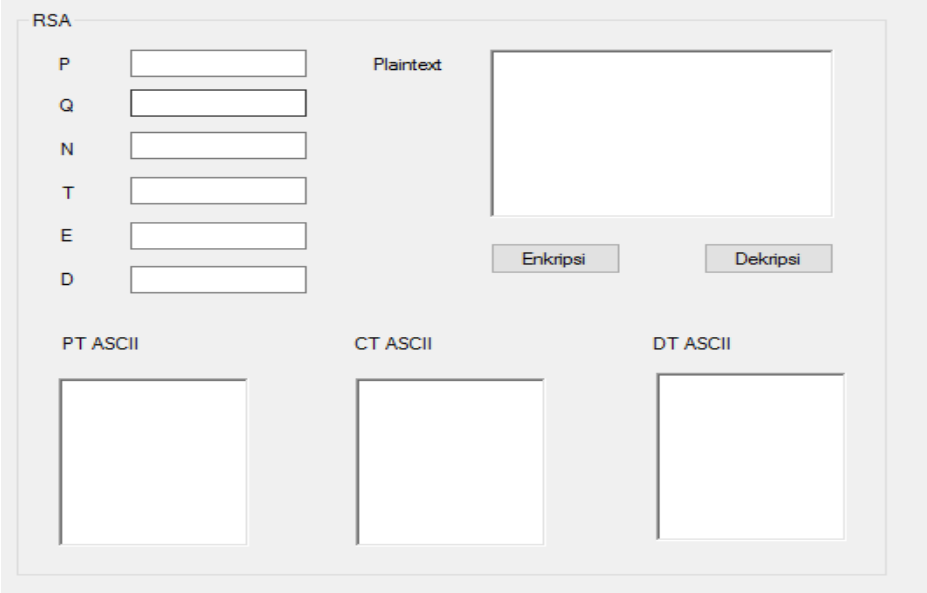
Didalam menu proses ini, terdapat 2 (dua) bagian pemrosesan yaitu proses pembangkitan kunci yang perhitungannya menggunakan metode *The Sieve Eratosthenes* dan proses dekripsi dan dekripsi yang perhitungannya menggunakan algoritma RSA.

Berikut merupakan bagian dari metode *The Sieve Eratosthenes* yaitu:



Gambar 4.2.4 Tampilan Halaman Proses Bagian SoE
Sumber : Hasil Percobaan Aplikasi (2019)

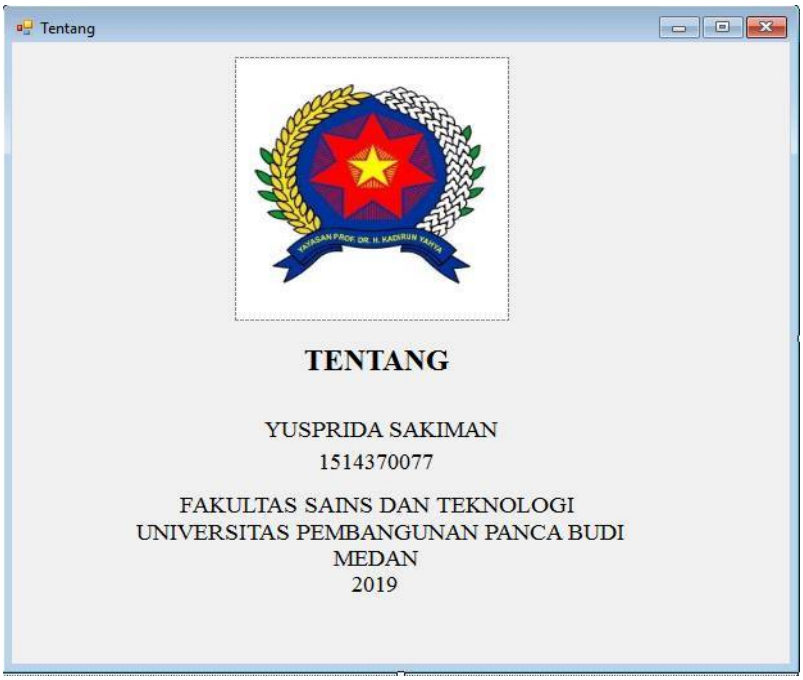
Berikut merupakan bagian dari metode *The Sieve Eratosthenes* yaitu:



Gambar 4.2.5 Tampilan Halaman Proses Bagian RSA
Sumber : Hasil Percobaan Aplikasi (2019)

4. Tampilan Halaman Tentang

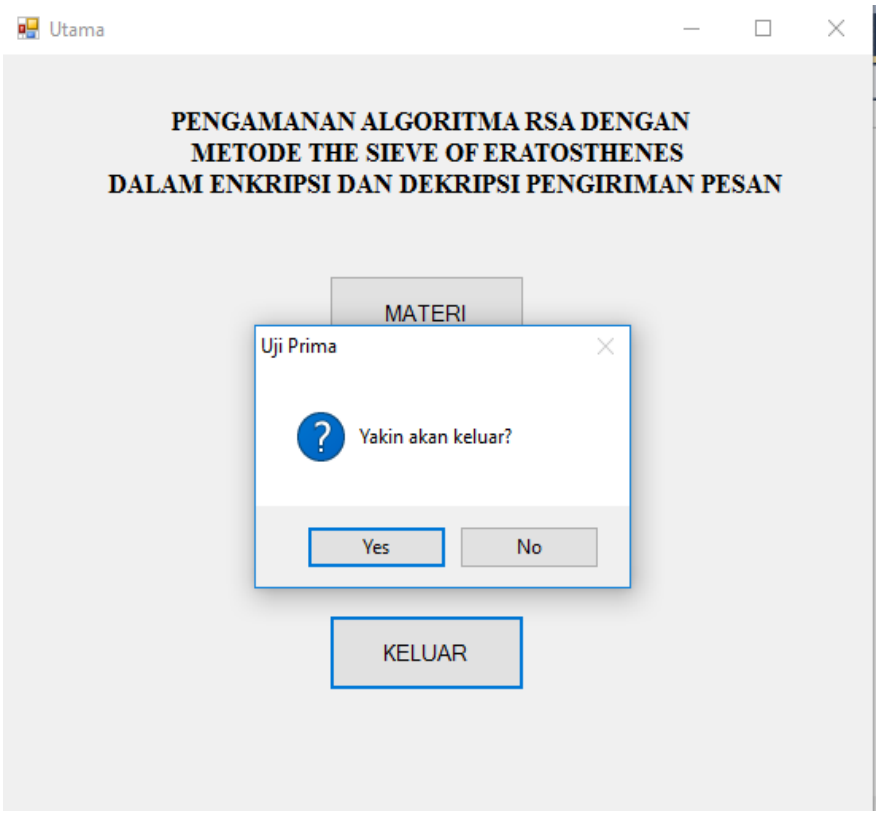
Tampilan berikut berisi tentang profil penulis.



Gambar 4.2.6 Tampilan Halaman Tentang
Sumber : Hasil Percobaan Aplikasi (2019)

5. Tampilan Halaman Keluar

Sebelum benar-benar keluar dari program, pengguna akan diingatkan kembali apakah ingin benar ingin keluar, atau terdapat kesalahan dalam penggunaan. Berikut merupakan tampilan dari menu keluar.



Gambar 4.2.7 Tampilan Halaman Keluar
Sumber : Hasil Percobaan Aplikasi (2019)

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan pembahasan yang telah diuraikan dibab-bab sebelumnya, maka penulis dapat menarik kesimpulan sebagai berikut:

1. Aplikasi yang dirancang menggabungkan antar algoritma RSA dengan metode *The sieve of Eratosthenes.*
2. Aplikasi yang dirancang akan digunakan untuk mengamankan isi pesan dari pihak yang tidak diinginkan.
3. Metode *The sieve of Eratosthenes* berfungsi untuk mengacak bilangan prima yang akan digunakan.

5.2 Saran

Berikut merupakan saran untuk melakukan pengembangan pada sistem selanjutnya ialah sebagai berikut:

1. Diperlukan penambahan range angka pada bilangan prima yang digunakan sehingga angka akan semakin berpariasi.
2. Aplikasi ini dihapapkan dapat dikembangkan menggunakan metode-metode lainnya.

3. Aplikasi ini diharapkan dapat dikembangkan lebih lanjut, sehingga dapat digunakan diandroid.

DAFTAR PUSTAKA

Albert Ginting, R. Rizal Isnanto, Ika Pertiwi Ikasari. (2015). Implementasi Algoritma Kriptografi RSA Untuk Enkripsi Dan Dekripsi Email. *Jurnal Teknologi Dan Sistem Komputer.* 3. 253-258.

Arif Suganda, Sinar Sinurat, Saidi Ramadan. (2018). Penerapan Algoritma Sieve Of Eratosthenes Untuk Pembangkit Bilangan Acak . *Jurnal Pelita Informatika*. 17. 382-385.

Dony Ariyus. (2006). Kriptografi, Keamanan Data Dan Komonikasi**.** Yogyakarta : Graha Ilmu.

Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. Int. J. Recent Trends Eng. Res, 3(7), 279-284.

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19.

Mariance, U. C. (2018). Analisa dan Perancangan Media Promosi dan Pemasaran Berbasis Web Menggunakan Work System Framework (Studi Kasus di Toko Mandiri Prabot Kota Medan). Jurnal Ilmiah Core IT: Community Research Information Technology, 6(1).

Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). Int. J. Eng. Trends Technol, 38(7), 380-383.

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Muttaqin, Muhammad. "Analisa pemanfaatan sistem informasi e-office pada universitas pembangunan panca budi medan dengan menggunakan metode utaut." Jurnal Teknik dan Informatika 5.1 (2018): 40-43.

M. Safri Lubis, M. Andri Budiman, Karina Lolo Manik. (2013). Penggunaan Algoritma RSA Dengan Metode The Sieve Of Eratosthenes Dalam Enkripsi Dan Dekripsi Pengiriman Pesan. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI).* 28-33.

Opik Taupik K, Mohamad Irfan, Ai Nurpianti. (2013). Pembuatan Aplikasi Anbiyapedia Ensiklopedi Muslim Anak Berbasis Web. *Jurusan Teknik Informatika*. 7 .37-38.

Prabowo Pudjo Widodo, Herlawati. (2011). Menggunakan UML. Bandung : Informatika.

Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 8-18.

Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN. 2015.

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." INTECOMS: Journal of Information Technology and Computer Science 1.1 (2018): 72-77.

Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. International Journal of Recent Trends in Engineering & Research, 3(8), 196-201.

Rahim, R. (2018, October). A Novelty Once Methode Power System Policies Based On SCS (Solar Cell System). In International Conference of ASEAN Prespective and Policy (ICAP) (Vol. 1, No. 1, pp. 195-198).

Rio Irawan, Ilhamsyah, Yulrio Brianorman. (2015). Aplikasi Enkripsi Dan Dekripsi Pesan Singkat Menggunakan Algoritma Knapsack Berbasis Android. *Jurnal Coding Sistem Komputer Untan.* 3. 57-66.

Rolly Yesputra. (2017). Belajar Visual Basic. Net Dengan Visual Studio 2010. Medan : Royal Asahan Press.

Rosa A.S, M.Shalahuddin. (2013). Rekayasa Perangkat Lunak. Bandung : Informatika.

Sentot Kromodimoeljo. (2010). Teori dan Aplikasi Kriptografi. SPK IT Consulting.

Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Int. J. Sci. Res. Sci. Technol, 3(6), 470-473.