PENERAPAN ALGORITMA RSA UNTUK KEAMANAN PESAN INSTAN PADA PERANGKAT ANDROID

Muhammad Arif Zainuddin[1)], Dadang Iskandar Mulyana[2)]
[1]Teknik Informatika, STIKOM CKI
Email: marifzainuddin@gmail.com
[2]Teknik Informatika, STIKOM CKI
Email: fokus2008@yahoo.com

Abstract: *This time the use of smart phone is very popular among every circle of society, especially Android base device. Start form low end society to mid high society are so familiar with Android base smartphone. One of feature that is use by many Android users is Instant Messenger. Instant Messenger is favored because it speed to transmit message and cost that relatively low, it count base on size of data that transmitted. As many user that use this service secrecy feature to ensure safety of text that is send is became main attention of users. Cryptography is one solution that can be use and develop to keep secrecy of instant messenger. With encryption, secrecy and safety of text can be made. Cryptography have many technique to do encryption of text or message, one of those technique that have high security and hard to decode is RSA algorithm. Implementation of RSA algorithm to Instant Messenger application is to encrypt message using public key before message is transmit. On receiver side using private key that generated to decrypt received message. Purpose of this this is to make sure message can only be read by intended receiver. If message is receive by unintended receiver then send information will be hard to understand, because message that is received only show number of codes that hard to described. Message that is send is receive by server application and be continued to intended receiver.*

Keywords: Android, Instant Messenger, RSA

1. PENDAHULUAN

Android adalah suatu sistem yang di gunakan alat komunikasi telepon pintar yang perkembangannya cukup pesat. Android bisa di aplikasikan pada telepon pintar dengan segmen harga yang cukup luas, dari yang murah hingga yang mahal. Hal inilah yang menjadikan Android memiliki pasar yang luas.

Pesan instan adalah suatu aplikasi pengirim pesan cepat dengan perantara jarinngan internet yang ada pada perangkat Android. Pesan instan dipilih kerena kecepatan pengiriman data dan murahnya biaya yang digunakan, dalam mengirim sebuah pesan, biaya dihitung berdasarkan besarnya data yang dikirim atau diterima.

Karena menggunakan jaringan internet, menjaga kerahasiaan pesan dianggap penting. Salah satu cara untuk menjaga kerahasian data adalah menggunakan kriptografi. Prinsip dasar kriptografi adalah proses enkripsi dan dekripsi. Enkripsi adalah proses mengubah plain text (text yang bisa dimengerti) menjadi chipher text (text yang sulit dimengerti), sedangkan dekripsi adalah kebalikan dari enkripsi.

Salah satu algoritma kriptografi yang banyak digunakan adalah RSA. Algoritma RSA dibuat oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1976. Menggunakan kunci pubik dan kunci privat untuk enkripsi dan dekripsi pesan. Salah satu keunggulan pada algortima RSA terletak pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor prima.

Berdasarkan hal ini peneliti ingin membangun sebuah aplikasi pesan instan pada perangkat Andrtoid yang menginplementasikan algoritma RSA dalam proses enkripsi dan dekripsi pesan teks.

2. METODE PENELITIAN

Pada tahun 2012 N. Saravan, A.Mahendiran, N.Venkata Subramanian dan N.Sairam melakukan penelitian menggunakan algoritma RSA pada lingkungan cloud. Tujuan penelitian ini adalah melindugi data yang tersimpan pada cloud menggunakan keamanan algoritma.

Algoritma RSA di implementasikan pada pada Google App menggunakan Cloud SQL. “Dari hasil pengujian diketahui bahwa RSA memberikan perlindungan pada data yang disimpan pada cloud SQL” (N. Saravan. Dkk, 2012).

Parsi Kalpana dan Sudha Singaraju pada tahun yang sama juga menggunakan algoritma RSA untuk mengenkripsi data untuk menyediakan kemanan pada cloud, sehingga hanya pengguna yang dipilih yang bisa mengakses data. Sebelum data disimpan data di enkripsi terlebih dahulu dengan algoritma kemudian disimpan pada cloud. Pada saat dibutuhkan, pengguna meminta data pada penyedia cloud, penyedia cloud mengenali user kemudian mengirimkan data kepada user.

Pada saat penyimpanan data penyedia cloud memberikan kunci publik untuk mengenkripsi data dan selanjutnya pengguna dipetakan dengan bilangan integer menggunakan protokol yang telah disepakati. Data yang telah di enkripsi kemudian disimpan pada cloud.

Saat pengguna membutuhkan data yang disimpan, penyedia cloud memberikan data yang terenkripsi

kepada pengguna. Selanjutnya pengguna kemudian mendekripsi data tersebut dennga kunci privat yang dimilikinya sehingga data dapat diterima.

Hanya user yang memiliki otoritas yang bisa mengakses data, walaupun secara kebetulan atau disengaja data dapat di lihat oleh pihak lain, pihak tersebut tidak bisa mendekript data tesebut sehingga keamanan data bisa tercipta. "Keamanan data dapat disediakan dengan implementasi algoritma RSA"(Parsi Kalpana dan Sudha Singaraju, 2012).

Busran dan Novernus Ayundha Putra pada 2014 menggunakan lagoritma RSA untuk bahasa pemograman JAVA demi menjaga keamanan file. Mengembangkan aplikasi berbasis desktop untuk melakukan fungsi enkripsi dan dekripsi file.

Proses enkripsi di lakukkan dengan pembuatan kunci publik dan kunci privat. Dengan kunci public yang didapatkan file dengan extensi *.txt di enkripsi dan menghasilkan file baru dengan extensi *.chipertext. Hal yang dibutuhkan pada saat dekripsi antara lain : file dengan extensi *.chipertext dan kunci privat yang dimiliki pengguna. Kunci ini selanjutnya di gunakan aplikasi untuk mendekripsi file.

Dari hasil penelitian yang di lakukan di ketahui bahwa "Lama waktu proses pengolahan sebuah informasi menjadi sebuah sandi dengan mengaplikasikan algoritma RSA berbasis pemrograman java sangat dipengaruhi oleh ukuran file yang akan diolah"(Busran, Novernus Ayundha Putra, 2014). Menggunakan algoritma RSA dalam penyandian file menggunakan program berbasis JAVA dapat membantu pengguna mengamankan file yang bersifat rahasia.

Anang Paramita Wahyadyatmika. Dkk 2014 menggunakan algortima RSA untuk pengamanan pesan pada pesan elektronik email. Sebelum pesan email di kirim, pesan tersebut harus di enkripsi oleh pengirim pesan dengan kunci publik berdasarkan data pemerina. Setelah pesan di enkripsi kemudian pesan di kirim kepada penerima. Penerima yang menerima pesan yang di enkripsi melakukan dekripsi menggunakan kunci privat yang dimilikinya.

Pada penelitian ini algortima RSA diuji terhadap beberapa serangan. Serangan tersebut antara lain serangan pada chiper text saja, serangan faktorisasi, dan serangan brute force. Dari hasil pengujian peneliti menyimpulkan bahwa "Algoritma kriptografi nirsimetri RSA sangat baik untuk mengatasi masalah manajemen distribusi kunci yaitu dengan menyimpan pasangan kunci pada basis data sedangkan kunci yang di distribusikan hanya kunci publik"(Anang Paramita Wahyadyatmika. Dkk, 2014). Algoritma RSA termasuk algoritma yang baik (secara komputasi). Dengan jumlah chipertext yang lebih panjang dari plaintext menyebabkan usaha untuk melakukan dekripsi dengan faktorisasi membutuhkan waktu yang lama.

3. DASAR TEORI

Kriptografi

Penggunaan kriptografi pada kelompok bertujuan untuk menjaga privasi dan keamanan informasi yang dikirim satu sama lain walaupun berada dalam jalur komuniksai yang sama dengan pihak lain. "Historically, cryptography arose as a means to enable parties to maintain privacy of the information they send to each other, even in the presence of an adversary with access to the communication"(**Mihir Bellare, Phillip Rogaway, 2005:7**).

Masalah terbesar kriptografi adalah memastikan keamanan informasi yang di kirim pada medium yang kurang aman. Sebagai contoh ada dua karakter yaitu pengirim S dan penerima R dan pihak lain adalah L. Tujuan kriptografi adalah agar informasi yang dikirim S hanya bisa diterima R, walaupun L juga memungkinkan melihat pesan yang dikirim tetapi informasi didalamnya tidak bisa dibaca oleh L. Hal ini dilakukan dengan mengenkripsi *plaintext* menjadi *chiphertext.* "Plaintext atau cleartext adalah pesan informasi yang dapat dibaca"(**Kurniawan Yusuf,2004:1**) sedangkan "Enkripsi adalah tehnik yang digunakan untuk memebuat pesan menjadi tidak bisa dibaca atau disebut chiphertext"(**Kurniawan Yusuf,2004:1**).

Aspek Kriptografi

Tujuan kriptografi adalah memberikan keamanan. Dalam memberikan layanan keamanan kriptografi harus meliputi aspek-aspek :

1. Authority, menjaga informasi dari pihak yang tidak memiliki otoritas.
2. Integrity, bahwa informasi yang yang di terima tidak berubah dan sesuai dengan aslinya.
3. Authentication, pengenalan pada pengguna yang saling berhubungan dan identifikasi kebenaran informasi.
4. Nonrepudation, bahwa penerima dan pengirim informasi tidak bisa memberikan penyangkalan atas informasi yang telah diterimanya.

Algoritma RSA

"Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (Massachussets Instittute of Technology) pada tahun 1976. Ketiga peneliti tersebut adalah Ron Riverst, Adi Shamir, dan Leonard Adleman. Algoritma RSA digunakan untuk membangkitkan 2 kunci yaitu kunci publik dan kunci privat. Kunci publik adalah dua buah variable bilangan (*e*,*n*) yang di gunakan untuk mengenkripsi data. Sedangkan kunci privat adalah dua buah variabel bilangan (*d*,*n*) yang di gunakan untuk melakukan dekripsi data. Nilai *e*, *d* dan *n* adalah nilai bilangan bulat positif.

Keamanan algoritma RSA bergantung pada kesulitan memfaktorkan bilangan besar. “We have proposed a method for implementing a publik-key cryptosystem whose security rests in part on the difficulty of factoring large numbers”(**R.L. Rivest, A. Shamir, and L. Adleman, 1977**).

Keamanan kunci privat dan kunci pulik pada algoritma RSA sangat bergantung pada dua variabel p dan q dimana variable ini di gunakan untuk menciptakan kedua kunci tersebut.

Berikut adalah proses dimana kunci publik dan kunci privat diciptakan.

a) Tentukan dua nilai p dan q, dimana p dan q adalah bilangan prima contoh : $p = 3\ dan\ q = 11$.
b) Hitung nilai n dari hasil p . q, $n = p.q = (3)(11) = 33$ .Maka nilai n = 33.
c) Hitung nilai $\emptyset(n) = (p-1)(q-1) = (3-1)(11-1) = (2)(10) = 20$.
d) Pilih nilai e dimana $1 < e < \emptyset(n)$ dan e adalah nilai prima, ditentukan nilai $e = 7$.
e) Hitung nilai d dimana $(d.e)\ mod\ \emptyset(n) = 1$, salah satu solusinya adalah $d = 3$ perhitungannya $((3)(7))\ mod\ 20 = 1$.
f) Karena nilai d, e, dan n telah diketahui maka kunci publik adalah (e,n) atau (7, 33) dan kunci privat adalah (d,n) atau (3,33).
g) Proses enkripsi bilangan $m = 2$ => $c = 2^7\ mod\ 33 = 29$.
h) Proses dekripsi nilai $c = 29$ => $m = 29^3\ mod\ 33 = 2$.

Karena kunci privat d berisfat rahasia, maka semua element pembentuk nilai d juga harus rahasia, yaitu e dan $\emptyset(n)$. Namun nilai e bersifat publik yang di gunakan untuk enkripsi, ini menyisakan $\emptyset(n)$ harus bersifat rahasia maka dari itu bilangan pembentuk n juga harus rahasia (p dan q). Nilai p dan q tidak boleh sama karena jika p dan q sama maka nilai ini dapat ditentukan hanya dengan mencari pemfaktor nilai n yang digunakan untuk enkripsi $\sqrt{n} = p = q$.

ASCII

ASCII (American Standard Code of Information Interchange) adalah sebuah standar internasional dalam kode huruf dan symbol seperti HEX dan UNICODE yang memetakan kode numerik yang mempresentasikan karakter seperti a~z atau karekter simbol seperti ‘@’. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebesar 7 bit, namun ASCII disimpan sebagai 8 bit dengan menambahkan nilai 0 sebagai nilai signifikan paling tinggi.

Encoding pada ASCII menggunakan 3 tipe bilangan bulat yaitu decimal (2^2), hexadecimal (2^{16})dan oktadesimal (2^8).

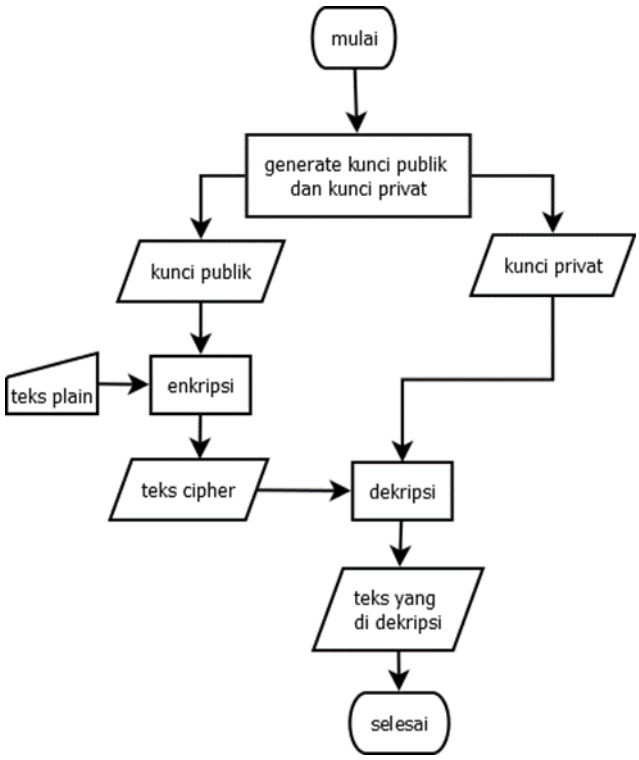
ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Gambar 2.1 Table ASCII

4. HASIL DAN PEMBAHASAN

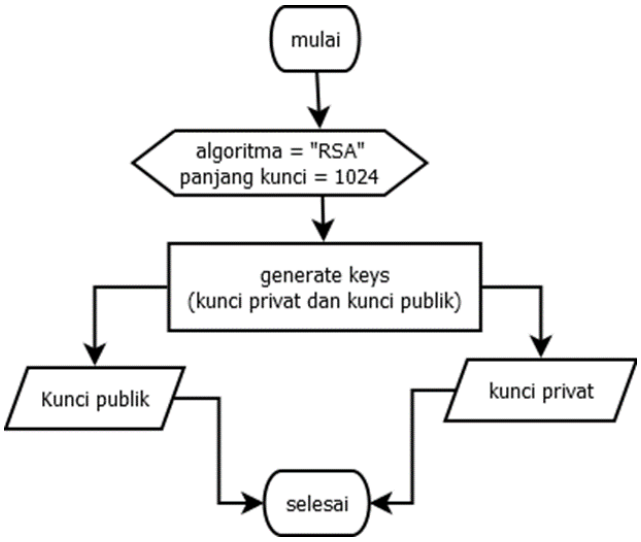
Data hasil penelitian yang dikumpulkan adalah data-data yang di gunakan dalam proses penelitian, serta pengujian algoritma RSA pada aplikasi pesan instan berbasis Android.



Gambar 3.1 Proses Enkripsi dan Dekripsi

Enkirpsi dan dekripsi pesan teks dimulai dari membuat kunci publik dan kunci privat. Kedua kunci ini memegang peranan penting dalam proses enkripsi dan dekripsi pesan teks.

4.1 Menciptakan Kunci Publik dan Privat



Gambar 3.2 Membuat Kunci Publik dan Kunci Privat

Kunci publik dan kunci privat di buat dengan menggunakan pustaka pada bahasa pemograman Java. Pustaka yang di gunakan adalah :

1. java.security.KeyPairGenerator.
2. java.security.KeyPair.
3. java.security.PublicKey.
4. java.security.PrivateKey.

Baris perintah untuk membuat kunci publik dan kunci privat adalah :

```
public void generateKey(){
    try{
        final KeyPairGenerator keygen = KeyPairGenerator.getInstance(“RSA”);
        keygen.initialize(1024);
        final KeyPair keypair= keygen.generateKeyPair();
        pbKey = keypair.getPublic();
        pvKey = keypair.getPrivate();
}catch(Exception e){
}
}
```

1. Dengan pustaka KeypairGenerator di tentukan jenis algoritma dan panjang kunci yang digunakan.
2. Pustaka KeyPair untuk menciptakan kunci publik dan kunci privat berdasarkan objek dari KeyPairGenerator

Hasil dari membuat kunci publik dan kunci privat adalah :

Kunci pulbik : OpenSSLRSAPublicKey{modulus=cbc80dc7a913231af24ecfb9e88dd766e52dfdf29f791b0ab63576b8357f6bb73ada9d99c27f14327b4933ac459d0f1f75e52ae861c5a252b5cfce3370d30c5d95be93f70dbe62b026f160b042acf80fa31077cfcfc19adf1baafc6a83336f8b256000dec75c6c51690da3865883588563b3991f02abc88155cbadb4e8b91527,publicExponent=10001}

Kunci Privat : OpenSSLRSAPrivateCrtKey{modulus=cbc80dc7a913231af24ecfb9e88dd766e52dfdf29f791b0ab63576b8357f6bb73ada9d99c27f14327b4933ac459d0f1f75e52ae861c5a252b5cfce3370d30c5d95be93f70dbe62b026f160b042acf80fa31077cfcfc19adf1baafc6a83336f8b256000dec75c6c51690da3865883588563b3991f02abc88155cbadb4e8b91527,publicExponent=10001,privateExponent=59c0ce18ef65e76359efce5c228a3ea22a34bc91dd1d5904b9c608790fcf04615a81a9426cc0cde3821b76afdca55560d4eb4f9fb45878ab173ae4a6117e53062794733ac93e1558b60633c5fdb1f84ca6ae12537ab5aa3ef7fc1b2e5287307f0067b53fa85e26e95e66f5dde747b39f014d94f3a2b8362f0dd358f641821b61,primeP=f070aff2c757486a5ab01bc2c62460b48b005008d73b9737e0d06f851e585a71b4f20b2d610a206e495d716cc3ea2f6c32cf5938b10d87a145c611ec5dba2db7,primeQ=d8f80cf1c77b41a02b1bbaa6c0c7a6227e379ce993aa5770d3810b53b099ac9d0ddac666a12f94c8e30a3d2da8857227664b74549d72c9b5f1b8e74e2a845411,primeExponentP=70e809c790d22ce03c7bcc5d775c27c94028c26c945d98521610eafd70d57e8b3cb4188993b304ada567ead66f5d6e2d79e2a27c1bb045cb768f5f654652221b,primeExponentQ=1a3938cf4c2df08b9c4a38b008e2d88898babe03592ea06ce993523c263f1ca6cad2e361ea1f671b349dbb31368a1277029d220d4c0e60a4d5f76435855c0311,crtCoefficient=87ae2f6b706173e7a6fccb34ec27c4a5f9a3f588e42a86011b3a6af2cf7df016cce7661120361bd7ad7f77170d681bcfa1178453b01502ac183a16d5f1f1ca7b

4.2 Pengumpulan dan Pengujian Data

Pengumpulan data teks yang akan di lakukan sebagai salah satu variabel dalam proses pengujian enkripsi dan dekripsi.

Tabel 4.1 Tabel Data Teks Yang Akan Diuji

No	Teks	Panjang Teks
1	percobaan teks pertama	22
2	teks dengan ukuran sedang	25
3	coba dengan teks yang panjang sekali yaitu teks yang panjang sekali	67
4	coba dengan emoticon ❖❖❖❖❖❖❖	33
5	teks panjang sekali teks panjang sekali teks panjang sekali teks panjang teks panjang sekali teks panjang sekali teks panjang sekali teks panjang sekali	154
6	teks panjang sekali teks panjang sekali teks panjang sekali teks panjang teks panjang sekali teks panjang sekali teks panjang	127
7	dengan karakter spesial $#&#;@)(%!$-@	38
8	karakter dengan angka 17384028	30

4.3 Proses Data Pengujian Algoritma RSA

Dalam pengujian setiap proses dicatat pada log aplikasi, berdasarkan log ini maka diketahui variabel yang akan diuji. Variabel tersebut adalah :

1. Plain teks: yaitu teks yang akan di enkripsi, teks ini di dapat berdasarkan maukkan data dari user.
2. Panjang teks: pajang teks adalah banyaknya karakter pada teks.
3. Waktu enkripsi: waktu enkripsi di dapatkan dengan mencatat log sebelum dan setelah proses enkripsi, selisih dari kedua waktu yang dicatat ini yang di jadikan acuan sebagai waktu enkripsi.
4. Teks cipher: adalah teks yang di hasilkan setelah proses enkripsi.
5. Waktu dekripsi: adalah waktu yang di butuhkan untuk proses dekripsi. Seperti waktu enkripsi, waktu dekripsi didapatkan dari selisih waktu pada saat mulai proses dekripsi sampai selesai proses dekripsi.
6. Hasil dekripsi adalah hasil dari teks cipher yang di dekripsi.

Tabel 3.3 Data Pengujian

No	Plain Teks	PT	WE	Teks Cipher	PC	WD	Hasil Dekripsi
1	percobaan teks pertama	22	21ms	C0910EF3C697C56D2FE5F95C9611C7CBE681F0329F704BB2A7BE32BC8EAE3F5FDAE5CA3FF95C1D542E4B898C3EDC03DF68BB52B245180F7628A25BAC632CAE441AC0BE7DAFA7F70B5AE559F0412337583EB5F00CC0BE8FA926E38B05FE1E6D57655A8B5EBEEC891413FB2A99BA440	256	25ms	percobaan teks pertama

				25708D973EFBEEAF083E81F85019A652356			
2	teks dengan ukuran sedang	25	25ms	56071AECE918E2316EAE6CA177067CB0577010DE9D7474891431F546FEAF4DD607E2712234F62DB702579E61D886BA69DCC5D48C4DCF4948EEBDBBEE3B47D5B4D13DDBA477317A77E5F7772126723196DA2D6F5E4FF0340F9DC60C1323C362E3C1918519B9208D947B81D1CB29202A29F5	256	31ms	teks dengan ukuran sedang
				BE68E6B303885BC70ED4581F46CD47			

Keterangan :

Plain Teks = Teks yang akan dilakukan enkripsi.

P T = Panjang dari plain teks.

W E = Waktu Enkripsi, waktu yang digunakan pada proses enkripsi.

Teks Cipher = Cipher teks hasil enkripsi.

P C = Panjang dari teks cipher.

W D = Waktu dekripsi, waktu yang digunakan pada proses dekripsi.

Hasil Dekripsi = Teks hasil dekripsi teks cipher.

No	Plain Teks	P T	W E	Teks Cipher	P C	W D	Hasil Dekripsi
3	coba dengan teks yang panjang sekali yaitu teks yang panjang sekali	67	7ms	63810CE4AACC681A421111ED16D30A719F8AD393C1AED535CF45522406F22D234B5EA027D571165C7F116A1AE0BFE356504EA5F06689A523958F9E338744887E8E8	256	24ms	teks dengan ukuran agak panjang dan ini adalah panjang

				9962F3 E78010 EBDF7 604BA8 4B702B 5B5085 581F6F CDC08 B692C7 ADDA7 5440576 405E78 CF74E4 324C62 D6A129 4B18AE 6191884 874F33 2F9F78 8B374E D957B8			
4	coba dengan emoticon ▯▯ ▯▯ ▯▯	33	9ms	A61001 49EF30 81186B 9E9566 9A957D BA846 AACC6 EBC058 69CE0B 31D872 40B3B3 E0ACF 006EB6 525488 E624A4 173EEC 0F317B CF170B 3D6994 FEA15 AAFD3 D089F2 A52F0F B0891F	256	21 ms	ciba dengan teks yang panjang sekali yaitu teks yang panjang sekali yaitu teks yang panjang sekali
				C3C623 9B6718 8EEF73 3291D7 C8DE26 DF4706 F018C6 8582E1 32AB94 BA6B2 67D898 3CB93 D443E2 E86EF0 412A51 5E2CFC 3ED85 A47D13 DF4466 235594			

Keterangan :

Plain Teks = Teks yang akan dilakukan enkripsi.

P T = Panjang dari plain teks.

W E = Waktu Enkripsi, waktu yang digunakan pada proses enkripsi.

Teks Cipher = Cipher teks hasil enkripsi.

P C = Panjang dari teks cipher.

W D = Waktu dekripsi, waktu yang digunakan pada proses dekripsi.

Hasil Dekripsi = Teks hasil dekripsi teks cipher.

No	Plain Teks	P T	W E	Teks Cipher	P C	W D	Hasil Dekripsi
5	teks panjang sekali teks panjang	154	NA	NA	NA	NA	NA

	sekali teks panjang sekali teks panjang teks panjang sekali teks panjang sekali teks panjang sekali teks panjang sekali						
6	teks panjang sekali teks panjang sekali teks panjang sekali teks panjang teks panjang sekali teks panjang sekali	127	10ms	51E9199025563 0DD62F685039 A0CDD30B3FC 0C58DAE7CD BF4AE021C59 6312929 00535E5D5B71 DCAD8757E75 A03266BA31A 16B6B15D8A49 6129DBDFA464 CEEAA	256	24ms	teks panjang sekali teks panjang sekali teks panjang sekali teks panjang teks panjang sekali teks panjang sekali

	teks panjang			98B3506FFC125F83F93FFCC542BD46E63A64DDA0E7B6BC97A6CCCFBFD9A3A9FB80F6744A553697FCE3162BD73C3AAFEEEE8A9B7802B3ABF2369E1D355D333F7F4			

Keterangan :

Plain Teks = Teks yang akan dilakukan enkripsi.

P T = Panjang dari plain teks.

W E = Waktu Enkripsi, waktu yang digunakan pada proses enkripsi.

Teks Cipher = Cipher teks hasil enkripsi.

P C = Panjang dari teks cipher.

W D = Waktu dekripsi, waktu yang digunakan pada proses dekripsi.

Hasil Dekripsi = Teks hasil dekripsi teks cipher.

No	Plain Teks	P T	W E	Teks Cipher	P C	W D	Hasil Dekripsi
7	dengan karakter	38	7ms	628D075C16EC7AD82922E9B9	256	29ms	dengan karakter spesial $#&#;@

	spesial $#&#; @)(%! $-@			F08CC9 ED1EE 4F7CD B7BA6 E44FD BA4FC 619135 BE90D BCB173 92A7A9 5C0A93 62B02B E9A692 0A0B0 BC7227 486A30 5D3B48 6181D4 DD5AC 1FF3BB D53D87 1836313 2A51D A37257 EF2077 5CD3B 71AF34 CC18B3 75C549 DDFE6 AFC37 B851C D002A C4B636 1DD166 6CA449 6EDEF6 7DD10 A2D0D 6E28DC 517375 B			)(%!$- @

8	karakter dengan angka 17384028	30	9ms	62EF0C 61CC42 5BB936 ECC968 392B6B 7B1AD 0602F6 27A2F9 AA0467 C2F5E D8C11 C0768B 58B9D A495B BE67C BBDB0 B130E9 D262F4 B19E08 0457DA 7320D8 E68B1B F5074A 74740A C37160 AB540 D94453 365735 A65B28 59D8E2 9447DE D281EE 36689B F19D08 BBBF6 EB4CA 61FAE3 28B85B 95835A 0B8051 20FCD8 B9EC00 5BF5F4 09FD37 08	256	35 ms	mencoba dengan karakter spesial #-;:)#(*;#-{¥}•Δ`°°€£

4.4 Hasil Pengujian

Perbandingan panjang teks dengan, waktu enkripsi serta waktu dekripsi :

No	1	2	3	4	5	6	7	
Panjang Teks	25	25	67	33	54	27	38	0
Waktu Enkripsi	21ms	25 ms	7 ms	ms		0 ms	7 ms	ms
Waktu Enkripsi	25 ms	31 ms	24 ms	1 ms		4 ms	29 ms	5 ms

Tabel 3.4 Tabel Panjang Teks, Waktu Proses Enkripsi dan Dekripsi

Rata-rata

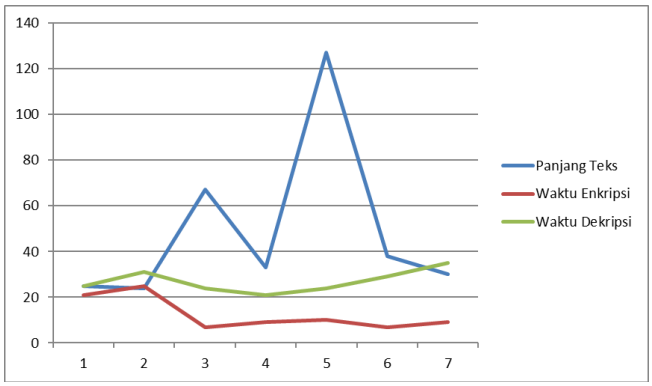
Nilai rata-rata dihitung hanya pada sampel yang berhasil dilakukan enkripsi dan dekripsi :

N = 8

Rata-rata panjang teks = $\frac{\sum Panjang\ Teks}{N} = \frac{25+25+67+33+127+38}{8} = 39.375$

Rata-rata waktu enkripsi = $\frac{\sum waktu\ enkripsi}{N} = \frac{21+25+7+9+10+7+9}{8} = 11ms$

Rata-rata waktu dekripsi = $\frac{\sum waktu\ dekripsi}{N} = \frac{25+31+24+21+24+29+35}{8} = 23.625ms$



Gambar 3.1 Diagram Relasi Panjang Teks, Waktu Enkripsi Dan Waktu Dekripsi

5. KESIMPULAN

Berdasarkan pengujian dan pengolahan data maka dapat disimpulkan:

1. Algoritma RSA adalah algoritma asimetris yaitu algoritma yang mempunyai dua kunci berbeda untuk proses enkripsi dan dekripsi yaitu kunci publik dan kunci privat.
2. Kunci publik di gunakan untuk mendekripsi teks biasa menjadi teks yang terenkripsi atau disebut teks cipher.
3. Kunci privat digunakan untuk mengembalikan teks cipher menjadi teks biasa.
4. Penggunaan algoritma RSA untuk aplikasi pesan instan adalah untuk mengenkripsi pesan sebelum dikirim kepada penerima.
5. Untuk kebutuhan enkripsi, pengirim pesan meminta kunci publik dari penerima yang di distribusikan secara bebas.
6. Pesan yang diterima kemudian di dekripsi menggunkan kunci privat yang hanya dimiliki oleh penerima pesan.

6. REFERENSI

R.L. Rivest, A. Shamir, and L. Adleman. *A Method For Obtaining Digital Signatures and Public-Key Cryptosystems.* Laboratory for Computer Science, Massachusetts Institute of Technology.Cambridge.1977.

Mihir Bellare, Phillip Rogaway. *Introduction To Modern Cryptography*. ©Mihir Bellare and Phillip Rogaway.1997–2005.

N. Saravanan, A. Mahendiran, N. Venkata Subramanian and N. Sairam. *An Implementation of RSA Algorithmin Google Cloud Using Cloud SQL*, ISSN: 2040-7467.2012.

Parsi Kalpana, Sudha Singaraju. *Data Security In Cloud Computing Using RSA Algorithm*. ISSN 2278-5841, Vol 1, Issue 4, September 2012.

Anang Paramita Wahyadyatmika, R. Rizal Isnanto, and Maman Somantri. *Implementasi Algoritma Kriptografi RSA Pada Surat Elektronik (E-Mail)*, ISSN: 2302-9927.2014.

Busran, Novernus Ayundha Putra. *Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma RSA Pada Sistem Keamanan File Berbasis Java*. ISSN: 2338-2724.2014.

Hariyanto,Bambang. *Esensi-esensi Bahasa Pemograman Java.* Bandung. INFORMATIKA.2011.

Kadir,Abdul. *Tuntunan Praktis: Belajar Database Menggunakan MySQL*. Yogyakarta. ANDI.2008.

Kadir Abdul. *Dasar Pemograman & Implementasi Database Relasional*. Yogyakarta. ANDI.2008.

Kadir Abdul. *Algoritma Pemograman Menggunakan C & C++*. Yogyakarta.Andi.2012.

Puji Agus Kurniawan. *Sistem Informasi Manajemen*. Semarang. CV Agung.1998.