Perbandingan Efisiensi Algoritma RSA dan RSA-CRT Dengan Data Teks Berukuran Besar

Sulistiyorini[1], Agus Prihanto[2],

[1] Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

[2]Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

[1]sulistiyorinisulistiyorini@mhs.unesa.ac.id

[2]agusprihanto@unesa.ac.id

***Abstrak*— RSA (Rivest Shamir Adleman) merupakan algoritma asimetris yang paling banyak digunakan dalam kriptografi. Dalam beberapa penelitian menyebutkan bahwa proses RSA memakan waktu yang cukup lama. Untuk mempercepat waktu proses dari RSA dilakukan penambahan suatu algoritma CRT (Chinese Remainder Theorem) untuk mengurangi perhitungan aritmatika modular dengan modulus besar dalam RSA yang disebut RSA-CRT. Dalam penelitian ini penulis akan membandingkan dua algoritma asimetris yaitu RSA (Rivest Shamir Adleman) dengan modifkasi RSA yaitu RSA-CRT (Rivest Shamir Adleman-Chinese Remainder Theorem) pada data teks berukuran besar dari segi efisiensi waktu. Algoritma RSA dan RSA-CRT digunakan untuk mengenkripsi dan mendekripsi suatu data teks berukuran 5 mb, 10 mb, 15 mb dan 20 mb untuk dibandingkan efisiensi waktu atau segi kecepatan prosesnya. Hasil pengujian waktu dari penilitian ini menunjukkan bahwa nilai waktu dari proses enkripsi antara algoritma RSA dan RSA-CRT tergolong sama sebab kunci publik yang digunakan rumusnya memang sama. Sedangkan dari proses dekripsi menunjukkan bahwa algoritma RSA-CRT lebih cepat dari pada algoritma RSA biasa. Dari pengujian kecocokan kunci menunjukkan bahwa kunci yang bangkitkan dan digunakan untuk proses enkripsi dekripsi harus berpasangan antara kunci publik dan kunci privatnya. Hasil pengujian ukuran file dari proses enkripsi mengalami kenaikan dengan rata-rata 42.757 kb, sedangkan dari proses dekripsi berkurang dengan rata-rata 0,007 kb dikarenakan terdapat karakter yang tidak dikenali sistem yaitu simbol bullets list yang mengakibatkan file tidak dapat kembali 100% seperti semula. Sehingga dapat disimpulkan bahwa dalam proses dekripsi suatu data teks berukuran besar algoritma RSA-CRT memiliki efisiensi waktu 50% dibandingkan dengan algoritma RSA biasa.**

***Kata Kunci*— Algoritma Elgamal, Kriptografi, Enkripsi, Dekripsi, Gambar Warna.**

I. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi pada era modern ini memiliki pengaruh yang cukup besar dalam perananan kehidupan manusia. Salah satunya dalam bidang komunikasi. Teknologi telah merubah cara masyarakat dalam berkomunikasi. Jika masyarakat zaman dahulu untuk berkomunikasi jarak jauh masih menggunakan cara konvensional contohnya surat menyurat. Komunikasi di era teknologi kini menjadi lebih mudah dengan menggunakan bermacam alat komunikasi dari *hardware* maupun *software*. Komunikasi yang sering ditemui pada era ini adalah komunikasi berbasis teks. Komunikasi pesan teks dilakukan dengan mengirim pesan dari pengirim dan diterima oleh penerima pesan. Berbagai macam fasilitas yang dapat digunakan sebagai alat untuk komunikasi berbasis teks diantaranya email, sms, *chatting* dan masih banyak yang lain.

Dalam komunikasi salah satunya pengiriman pesan beberapa hal harus diperhatikan, yakni kerahasiaan (keamanan), kesatuan pesan, identitas pesan dan nirpenyangkalan. Oleh sebab itu dalam pengiriman pesan dibutuhkan suatu kunci atau sandi untuk pesan sebelum dikirimkan kepada penerima agar kerahasiaan pesan tetap terjaga dan integritas pesan tidak mudah dirubah oleh orang lain. Kriptografi adalah ilmu yang mempelajari teknik matematika yang berhubungan dengan askpek keamanan informasi seperti kerahasiaan, integritas data, otentikasi entitas, dan otentikasi asal [1]. Sedangkan langkah untuk mengamankan data dalam kriptografi disebut dengan algoritma kriptografi.

Diketahui ada dua algoritma kriptografi yakni algoritma simetris dan algoritma asimetris. Algoritma simetris memakai satu kunci untuk enkripsi dekripsinya, sedangkan algoritma asimetris memakai dua kunci untuk proses enkripsi dan dekripsinya, dengan kunci publik untuk proses enkripsinya dan kunci privat untuk proses dekripsinya. Salah satu algoritma asimetris yang banyak digunakan dalam kriptografi adalah RSA (Rivest Shamir Adleman). RSA ditemukan oleh para peneliti dari MIT (Massachussets Institute of Technology), yaitu Ronald Linn Rivest, Adi Shamir, dan Len Adleman pada tahun 1977 [2]. Sulitnya memfaktorkan bilangan yang begitu besar menjadi faktor-faktor prima adalah kunci keamanan dalam RSA. Bilangan yang cukup besar membuat proses semakin lama dan memperlambat proses dalam RSA.

CRT (*Chinese Remainder Theorem*) adalah algoritma untuk mengurangi kalkulasi dari aritmatika mudular dengan

menggunakan modulus besar untuk perhitungan yang sama untuk tiap-tiap faktor dari modulus [3]. Algoritma CRT dapat digunakan bersama algoritma RSA untuk memercepat proses yang disebut RSA-CRT.

Hasil penelitian yang dilakukan oleh (Chenchen Zang, 2016) menyatakan bahwa kombinasi RSA salah satu yang diusulkan adalah RSA berbasis CRT memiliki struktur yang lebih sederhana dan mudah untuk diimplementasikan. Kekuatan keamanan dalam enkripsi dan dekripsinya sama dengan RSA biasa namun lebih efisien dan hemat waktu dalam komputasi [4].

Penelitian tentang efisiensi waktu dari enkripsi dan dekripsi dari RSA telah dilakukan oleh Desi Wulansari dkk (Wulansari Desi dkk, 2016) objek dalam penelitian tersebut adalah proses implementasi RSA dengan nilai parameter n ukuran 1024 bit dan 2048 bit. Proses yang diamati yaitu kompeksitas waktu dari proses enkripsi dan dekripsi. Hasil penelitian menyatakan bahwa algoritma RSA tergolong membutuhkan waktu yg lama dalam proses komputasi [5].

Penelitian serupa juga dilakukan oleh Ashari Arief dkk (Arief, Ashari & Saputra, Ragil, 2016) dengan judul Implementasi Kriptografi Kunci Publik dengan Algoritma RSA-CRT pada Aplikasi Instan Messaging. Hasil penelitian menunjukkan proses dekripsi memakai algoritma RSA-CRT untuk 1.800 karakter dengan bit n dari 56 bit hingga 88 bit memiliki kecapatan rata-rata dua kali lebih cepat dibandingkan menggunakan algoritma RSA [6].

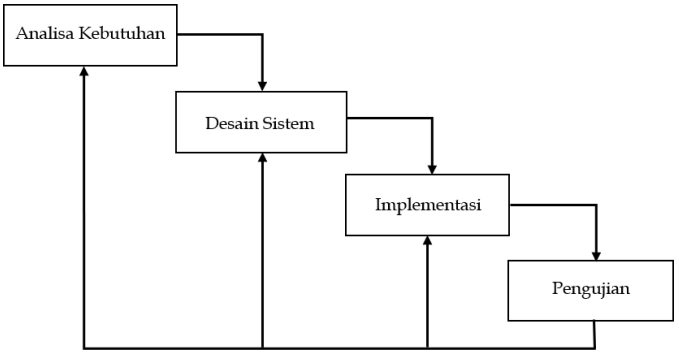
Penelitian tentang penerapan CRT telah dilakukan oleh Sura N. Abdulla dkk (Abdulla N. Sura & Al barak Alyaa , 2014) dalam penelitian menyebutkan bahwa CRT cocok untuk masalah keamanan dalam enkripsi dan dekripsi suatu pesan [7].

Penelitian yang dilakukan oleh Ari Muzakir dan Meigi Rahman (Muzakir, Ari & Rahman, Meigi, 2017) dengan judul Ujicoba Sistem Keamanan Informasi dengan Algoritma Kriptografi RSA-CRT pada Sistem E-Memo berbasis Mobile. Hasil penelitian menyebutkan bahwa algoritma kriptografi RSA diaplikasikan pada aplikasi e-memo untuk mengamankan suatu pesan pada e-memo tersebut, Algoritma RSA yang telah dimodifikasi menjadi RSA-CRT memiliki nilai untung dalam kecepatan proses dibandingkan dengan RSA biasa[8].

Berdasarkan hal tersebut, dilakukanlah penelitian untuk mengukur perbandingan efisiensi dari proses enkripsi dan dekripsi dilakukan modifikasi antara algoritma RSA dengan menggunakan CRT kemudian dibandingkan dari proses enkripsi dan dekripsi dari algoritma RSA biasa dengan menggunakan data teks berukuran besar. Maka perlu dibangun sebuah aplikasi dimana aplikasiini dapat melakukan enkripsi-dekripsi pada suatu pesan teks dengan menerapkan algoritma RSA-CRT untuk membandingkan efisiensi dari algoritma RSA dengan RSA- CRT.

II. METODOLOGI PENELITIAN

Berikut ini merupakan diagram alur dari penelitian yang telah dilakukan untuk proses "Perbandingan Efisiensi Algoritma RSA dan RSA-CRT dengan Data Teks Berukuran Besar" :



Gbr. 1 Metode Penelitian

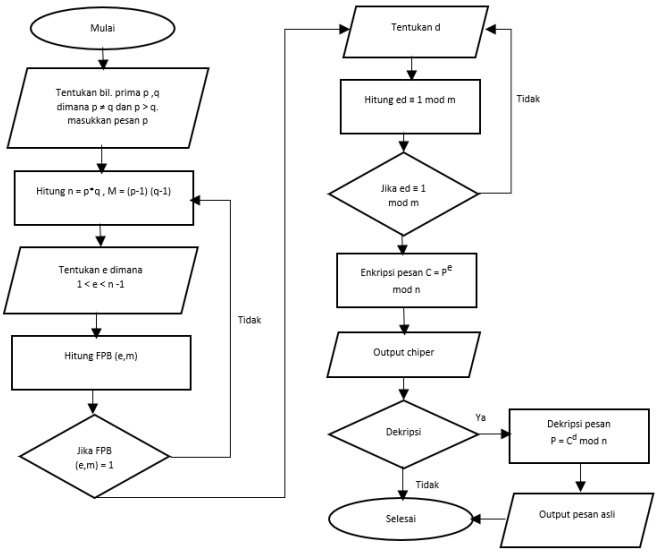
Dalam penelitian ini dilakukan perbandingan antara algoritma RSA dan algoritma RSA modifikasi CRT yaitu RSA-CRT dengan menggunakan data teks berukuran besar. Dengan tujuan untuk melihat perbandingan efisiensi waktu dari proses enkripsi dan dekripsi, perbandingan ukuran file hasil enkripsi dan dekripsi dan kecocokan kunci antara kedua algoritma tersebut. Tahapan proses yang dilakukan dalam penelitian ini yaitu :

A. ALGORITMA RSA

Dalam algoritma RSA terdapat 3 proses yaitu pembangkitan kunci, proses enkripsi dan dekripsi.

Pembangkitan Kunci

Proses pembangkitan kunci dilakukan untuk mendapatkan kunci publik dan kunci privat yang digunakan dalam proses enkripsi dan dekripsi. Flowchart algoritma RSA dapat dilihat pada Gbr. 2 berikut :



Gbr. 2 *flowchart* proses enkripsi dan dekripsi pada RSA

Langkah pembangkitan kunci RSA kunci :

1. Pilih dua buah bilangan prima sembarang p dan q dengan syarat p > q.
2. Hitung n = p * q.
3. Hitung m = (p-1) * (q-1).
4. Pilih sebuah bilangan bulat untuk kunci 86ystem, sebut 86ystem86 e, yang 86ystem86I prima terhadap m (86ystem86I prima berarti GCD(e, m) = 1) dengan syarat e ≠ (p-1), e ≠ (q-1), dan e < n.
5. Hitung kunci dekripsi, d, dengan kekongruenan ed ≡1 (mod m) atau d = e^{-1} mod m.

Proses Enkripsi

Setelah proses pembangkitan kunci selesai, kemudian beralih ke proses enkripsi dengan kunci publik yang dibangkitkan dengan menggunakan rumus (1) berikut:

$C = P^e \bmod n.$ (1)

Proses Dekripsi

Dekripsi digunakan untuk mengubah pesan hasil enkripsi (*ciphertext)* menjadi pesan asli (*plaintext)* yaitu dengan menggunakan rumus (2) untuk dekripsi RSA sebagai berikut:

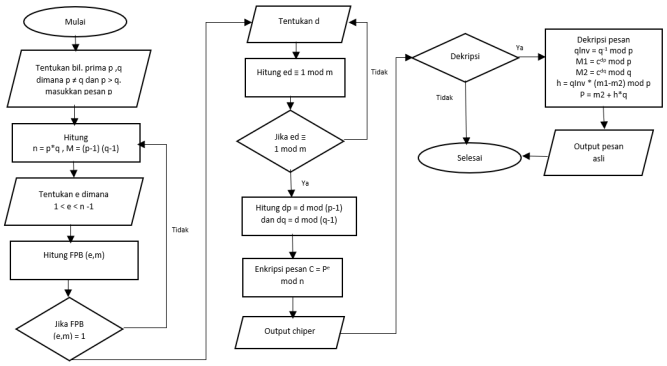
$P = C^d \bmod n.$ (2)

B. ALGORITMA RSA-CRT

Pada algoritma RSA-CRT juga terdapat 3 proses yaitu pembangkitan kunci, proses enkripsi dan dekripsi.

Pembangkitan Kunci

Proses pembangkitan kunci dilakukan untuk mendapatkan kunci publik dan kunci privat yang digunakan dalam proses enkripsi dan dekripsi. Flowchart algoritma RSA-CRT dapat dilihat pada Gbr. 3 berikut :



Gbr. 3 *Flowchart* proses enkripsi dan dekripsi pada RSA-CRT

Berikut merupakan langkah dari proses pembangkitan kunci RSA-CRT :

1. Misalkan p dan q adalah dua bilangan prima yang sangat besar dengan ukuran yang 86omput sama dimana p > q.
2. Hitung n = p*q dan m = (p-1)(q-1). Pilih sebuah bilangan bulat untuk kunci 86omput, sebut 86ompute e, yang 86omputer prima terhadap m (86omputer prima berarti GCD(e, m) = 1) dengan syarat e ≠ (p-1), e ≠ (q-1), dan e < n.
3. Hitung nilai d dengan rumus berikut.
   D = e^{-1} mod m
4. Hitung nilai dp dan dq dengan rumus berikut.
   Dp = d mod (p-1) dan dq = d mod (q-1)
   Kunci 86omput adalah <e> dan kunci rahasia adalah <d,dp,dq>.

Proses Enkripsi

Kunci publik RSA-CRT sama dengan sistem RSA yaitu (e) sehingga algoritma enkripsi tidak mengalami perubahan yaitu dengan menggunakan fungsi eksponensial modular yaitu seperti terlihat pada rumus (3) berikut.

$C = P^e \bmod n.$ (3)

Proses Dekripsi

Proses enkripsi dari algoritma RSA-CRT sama dengan proses enkripsi dari algoritma RSA standar. Jadi hanya akan difokuskan pada proses dekripsi RSA-CRT. Pesan didekripsi dengan menggunakan rumus $P = C^d \bmod n$ sehingga perhitungan tersebut tergantung pada nilai d dan n. Jika nilai dari d besar maka perhitungannya akan lebih lama sebab nilai eksponen yang besar d. Sedangkan pada RSA-CRT d akan digunakan untuk membangkitkan kunci dp dan dq dimana dp dan dq akan lebih kecil nilainya dari d sebab perhitungannya menggunakan d modulus p dan q. Rumus (4) berikut digunakan untuk menghitung dp dan dq.

$$\begin{aligned} &d \bmod (p-1) = e - 1 \bmod (p-1) \\ &d \bmod (q-1) = e - 1 \bmod (q-1) \\ &dp = e - 1 \bmod (p-1) = d \bmod (p-1) \\ &dq = e - 1 \bmod (q-1) = d \bmod (q-1) \end{aligned} \quad (4)$$

Dari hasil perhitungan di atas akan didapat nilai dp dan dq yang lebih kecil dari d. Selanjutnya akan dihitung representasi pesan m1 dan m2 yang akan digunakan untuk perhitungan akhir dari proses dekripsi dengan menggunakan rumus (5).

$$\begin{aligned} &M1 = c^{dp} \bmod p \\ &m2 = c^{dq} \bmod p \end{aligned} \quad (5)$$

m1 dan m2 dari hasil perhitungan di atas akan disubsitusikan ke dalam rumus Garner's (6) untuk menghitung solusi akhir dekripsi sebagai berikut:

$$\begin{aligned} &qInv = \left(\frac{1}{q}\right) mod\ p = 1 + \frac{kp}{q} \\ &h = qInv(m1-m2) \bmod p \\ &m = m2 + h.q \end{aligned} \quad (6)$$

III. HASIL DAN PEMBAHASAN

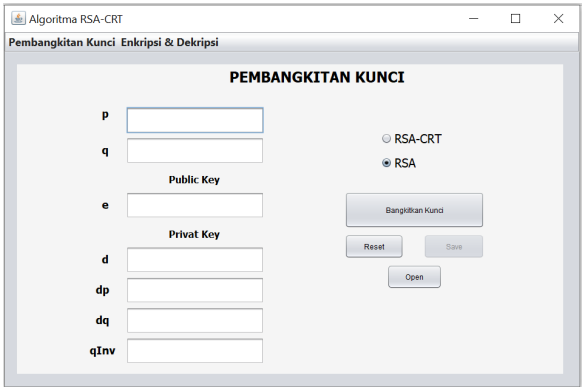
A. Data Penelitian

Dalam suatu penelitian agar memperoleh hasil yang maksimal maka diperlukan suatu data yang berkualitas dengan sumber yang terpercaya. Penelitian tugas akhir ini akan menggunakan data teks untuk bahan enkripsi dan dekripsi. Perangkat lunak yang dibangun akan digunakan untuk proses enkripsi dan dekripsi dengan menggunakan suatu data teks

berukuran besar. Data teks yang digunakan ada dua yaitu teks biasa berupa inputan langsung oleh pengguna pada suatu text field yang kemudian akan dieksekusi dan data teks atau file teks berukuran besar. Perangkat lunak tersebut akan menyuguhkan dua tampilan untuk proses enkripsi dan dekripsi yang pertama teks biasa dan kedua berupa file. Namun penelitian akan difokuskan pada proses enkripsi dan dekripsi suatu file data teks yang memiliki beberapa ukuran dengan membandingkan efisiensi dari hasil enkripsi dekripsi dua algoritma yakni RSA dan RSA-CRT sesuai dengan judul penelitian yang diusulkan sebelumnya. File data teks yang digunakan dalam penelitian ini yaitu beberapa file data teks berukuran besar yakni 5 mb, 10 mb, 15 mb dan 20 mb hasil dumb dari sql yang diperoleh dari dosen pembimbing skripsi penulis yakni Bapak Agus Prihanto S.T.,M.Kom.

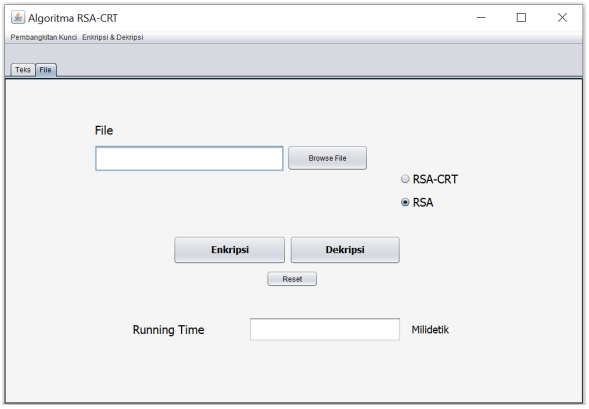
B. Desain dan Hasil Tampilan

Tampilan sistem ini dibangun dengan menggunakan *tool* java swing yang terdiri dari 2 halaman tampilan yaitu tampilan Pembangkitan Kunci dan tampilan Enkripsi & Dekripsi. Halaman tampilan pembangkitan kunci adalah halaman yang digunakan untuk melakukan proses pembangkitan kunci RSA dan RSA-CRT. Pada tampilan pembangkitan kunci akan ditampilkan beberapa textfield untuk nilai p, q, n, e, dp, dq, qlnv. Dan terdapat radio *button* untuk pilihan algoritma RSA dan RSA-CRT. Pada tampilan pembangkitan kunci ini juga terdapat 4 *button* yaitu *button* Bangkitan Kunci untuk proses pembangkitan kunci RSA dan RSA-CRT, *button* Save untuk menyimpan public key dan privat key pada lokasi penyimpanan 87ystem87I, *button* Open untuk membuka file kunci yang telah disimpan dan *button* Riset untuk mengosongkan field-field ketika selesai *digunakan*. Halaman tampilan pembangkitan kunci dapat dilihat pada Gbr 4.



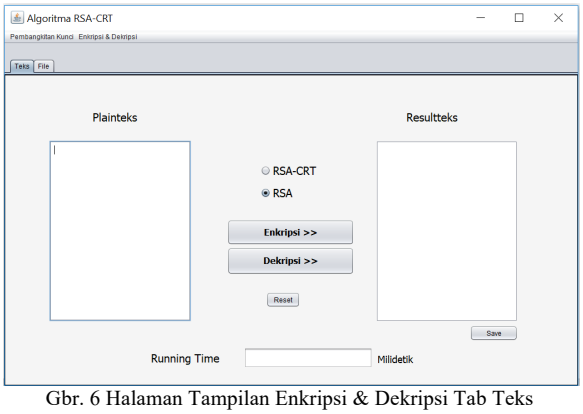
Gbr. 4 Halaman Pembangkitan Kunci

Halaman tampilan yang kedua yaitu halaman enkripsi dan dekripsi. Pada halaman ini terdapat 2 tab halaman lagi yakni file dan teks. Pada tab halaman file digunakan untuk melakukan proses enkripsi dan dekripsi file teks. Dalam halaman ini terdapat 2 *textfield* yang masing masing ialah file untuk inputan file yang akan dienkripsi dan *running time* untuk melihat hasil waktu dari proses. Kemudian terdapat 4 *button*, *button* browse untuk mencari file dari 87ystem87I yang akan proses, *button* enkripsi untuk mengenkripsi file, *button* dekripsi untuk mendekripsi file dan *button* reset. Pada halaman ini juga terdapat radio *button* yaitu RSA dan RSA-CRT untuk memilih salah satu algoritma yang akan digunakan dalam proses. Halaman tampilan enkripsi & dekripsi pada tab file dapat dilihat pada Gbr. 5.



Gbr. 5 Halaman Tampilan Enkripsi & Dekripsi Tab File

Halaman tampilan enkripsi & dekripsi selanjutnya yaitu tab teks. Pada halaman ini 87ystem sama dengan halaman tab file hanya saja halaman ini digunakan untuk proses enkripsi dan dekripsi suatu teks inputan dari pengguna pada text area. Pada tab teks ini terdapat 2 text area untuk menginputkan teks yaitu *plaintext* dan Resultteks, radio *button* untuk menentukan algoritma RSA dan RSA-CRT, text field untuk running time untuk melihat hasil waktu dari proses. Kemudian terdapat 5 *button* yakni *button* enkripsi untuk proses enkripsi, dekripsi untuk proses dekripsi, *button* save untuk menyimpan *plaintext* dan ciphertext dan *button* reset untuk mereset. Halaman tampilan enkripsi & dekripsi tab teks dapat dilihat pada Gbr. 6.



Gbr. 6 Halaman Tampilan Enkripsi & Dekripsi Tab Teks

Terdapat tujuh langkah yang harus dilakukan untuk melakukan proses enkripsi dan dekripsi.

1. Pengguna terlebih dahulu harus membangkitkan kunci pada halaman pembangkitan kunci.
2. Pengguna akan memilih algoritma yang akan dibangkitkan kuncinya, kemudian pengguna akan mengklik tombol Bangkitkan Kunci dan hasil nilai kunci

akan tampil pada *fieldtext*, kunci dapat disimpan dengan mengklik tombol *save.*

3. Pengguna akan melakukan enkripsi dan dekripsi teks biasa dengan menginputkan teks secara manual pada *textarea* Plainteks kemudian pengguna akan memilih algoritma sesuai kunci yang dibangkitkan, dan pengguna akan melakukan proses enkripsi dengan menekan tombol enkripsi. Hasil *ciphertext* akan tampil pada *resultext* dan lama proses akan tampil pada *running time*.
4. Untuk melakukan dekripsi pengguna akan menyalin *ciphertext* dari Resultteks pada Plainteks dan memilih algoritma sesuai proses enkripsi sebelumnya kemudian pengguna akan mengklik tombol dekripsi. Hasil dari dekripsi akan tampil pada Result teks dan lama proses dekripsi akan tampil pada *Running Time*.
5. Pada proses enkripsi dan dekripsi file pengguna akan menginputkan file dengan jenis txt, kemudian pengguna akan memilih algoritma sesuai kunci yang dibangkitkan.
6. Pengguna akan mengklik tombol enkripsi dan hasil dari enkripsi file akan tersimpan pada penyimpanan komputer lokal D folder KRIPTOGRAFI. Lama waktu dari proses enkripsi akan tampil pada *Running Time*.
7. Untuk melakukan proses dekripsi file, pengguna akan menginputkan file *ciphertext* jenis txt. Kemudian memilih algoritma sesuai proses enkripsi dan pengguna akan mengklik tombol dekripsi. Hasil dari proses dekripsi akan akan tersimpan pada penyimpanan komputer lokal D folder KRIPTOGRAFI. Lama waktu dari proses dekripsi akan tampil pada *Running Time.*

C. *Hasil Pengujian*

Dalam penelitian ini telah dilakukan 3 pengujian diantaranya pengujian waktu, pengujian kecocokan kunci dan pengujian ukuran file.

Hasil pengujian waktu dari proses enkripsi dan dekripsi dengan menggunakan algoritma RSA dan RSA-CRT dapat dilihat pada tabel berikut :

TABEL I
PENGUJIAN WAKTU ENKRIPSI RSA

No	Ukuran File	Pengujian Ke - (milidetik)					Rata-rata (milidetik)
		1	2	3	4	5	
1.	5 mb	12485	11777	9724	9114	10116	10643.2
2.	10 mb	18839	23434	22338	22011	22177	21759.8
3.	15 mb	35968	34362	58432	92610	59489	56172.2
4.	20 mb	84493	60351	42366	61645	58807	61532.4

TABEL II
PENGUJIAN WAKTU DEKRIPSI RSA

No	Ukuran File	Pengujian Ke – (milidetik)					Rata-rata (milidetik)
		1	2	3	4	5	
1.	5 mb	9371	9312	10850	9318	9598	9689.8
2.	10 mb	21829	19229	18249	20047	35124	22895.6
3.	15 mb	33940	32777	44838	28681	37645	35485.2
4.	20 mb	44045	51478	43512	41733	52314	46616.4

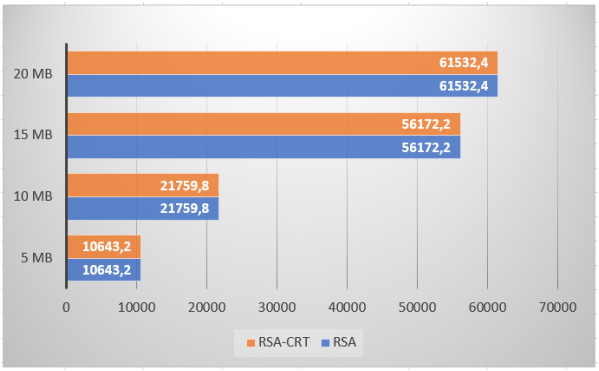
TABEL III
PENGUJIAN WAKTU DEKRIPSI RSA-CRT

No	Ukuran File	Pengujian Ke – (milidetik)					Rata-rata (milidetik)
		1	2	3	4	5	
1.	5 mb	12479	11783	9728	9110	10116	10643.2
2.	10 mb	18830	23434	22347	22020	22168	21759.8
3.	15 mb	35960	34370	58432	92619	59480	56172.2
4.	20 mb	84490	60351	42369	61638	58814	61532.4

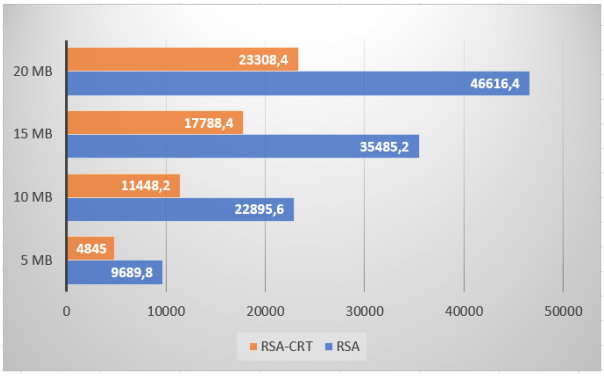
TABEL IV
PENGUJIAN WAKTU DEKRIPSI RSA-CRT

No	Ukuran File	Pengujian Ke – (milidetik)					Rata-rata (milidetik)
		1	2	3	4	5	
1.	5 mb	4686	4656	5425	4659	4799	4845
2.	10 mb	10915	9615	9125	10024	17562	11448.2
3.	15 mb	16970	16389	22419	14341	18823	17788.4
4.	20 mb	22023	25739	21756	20867	26157	23308.4

Untuk lebih memperjelas perbandingan efisiensi waktu antara algoritma RSA dengan algoritma RSA-CRT dapat dilihat grafik berikut.



Gbr. 4 Grafik Perbandingan Waktu dari Proses Enkripsi



Gbr. 4 Grafik Perbandingan Waktu dari Proses Dekripsi

Berdasarkan grafik di atas dapat dilihat bahwa proses enkripsi antara RSA dengan RSA-CRT 89ystem sama sedangkan proses dekripsi antara RSA dan RSA-CRT lebih cepat RSA-CRT sehingga dapat disimpulkan bahwa algoritma RSA-CRT memiliki efisiensi waktu 50% pada proses dekripsi 89ystem89ing dengan algoritma RSA biasa

Hasil pengujian kecocokan kunci pada masing-masing algoritma dilakukan dengan cara membangkitkan kunci yang menghasilkan kunci publik A dan kunci privat A. Kunci publik A digunakan untuk mengenkripsi teks "sembarangan". Kemudian dibangkitkan kunci lagi yang menghasilkan kunci publik B dan kunci privat B. *Chipertext* hasil enkripsi dengan kunci publik A akan didekripsi dengan kunci privat B. Hasil pengujian menyebutkan bahwa algoritma RSA maupun algoritma RSA-CRT kunci publik dan kunci privatnya harus berpasangan dalam melakukan proses enkripsi dan proses dekripsi. Jika tidak berpasangan maka proses enkripsi maupun dekripsi tidak bisa berjalan dengan baik seperti percobaan yang telah dilakukan.

Pengujian ukuran dari file yang telah dienkripsi dan didekripsi dengan algoritma RSA dan RSA-CRT dapat dilihat pada tabel berikut :

TABEL V
PENGUJIAN UKURAN FILE DARI ENKRIPSI ALGORITMA RSA

No	Ukuran	Enkripsi	Kenaikan
1.	5.005 KB	23.469 KB	18.464 KB
2.	10.010 KB	48.201 KB	38.191 KB
3.	15.014 KB	64.058 KB	49.044 KB
4.	20.019 KB	85.348 KB	65.329 KB

TABEL VI
PENGUJIAN UKURAN FILE DARI DEKRIPSI ALGORITMA RSA

No	Ukuran	Dekripsi	Penurunan
1.	5.005 KB	5.002 KB	0.003 KB
2.	10.010 KB	10.004 KB	0.006 KB
3.	15.014 KB	15.006 KB	0.008 KB
4.	20.019 KB	20.008 KB	0.011 KB

TABEL VII
PENGUJIAN UKURAN FILE DARI ENKRIPSI ALGORITMA RSA-CRT

No	Ukuran	Enkripsi	Kenaikan
1.	5.005 KB	23.468 KB	18.463 KB
2.	10.010 KB	48.202 KB	38.192 KB
3.	15.014 KB	64.063 KB	49.049 KB
4.	20.019 KB	85.343 KB	65.324 KB

TABEL VIII
PENGUJIAN UKURAN FILE DARI DEKRIPSI ALGORITMA RSA-CRT

No	Ukuran	Dekripsi	Penurunan
1.	5.005 KB	5.002 KB	0.003 KB
2.	10.010 KB	10.004 KB	0.006 KB
3.	15.014 KB	15.006 KB	0.008 KB
4.	20.019 KB	20.008 KB	0.011 KB

Berdasarkan keempat tabel di atas dapat disimpulkan bahwa ukuran file hasil enkripsi algoritma RSA dan RSA-CRT mengalami kenaikan rata-rata 42.757 KB. Sedangkan ukuran file hasil dekripsi hampir sama seperti ukuran awal sebelum proses enkripsi hanya berkurang dengan rata-rata 0,007 KB setelah didekripsi dikarenakan file yang dienkripsi dan didekripsi adalah file hasil dump sql. Terdapat karakter yang tidak dikenali oleh sistem yaitu simbol bullets list dan tidak terbaca sehingga ketika proses dekripsi ukuran file tidak dapat kembali 100% seperti semula.

IV. KESIMPULAN

Kesimpulan yang diperoleh berdasarkan percobaan yang dilakukan dalam penelitian ini menghasilkan sebuah System perangkat lunak yang dapat digunakan untuk proses enkripsi dan dekripsi dengan menggunakan dua algoritma yakni RSA (Rivest Shamir Adleman) standart dan RSA-CRT (Rivest Shamir Adleman – Chinese Remainder Theorem). Dalam uji coba yang telah dilakukan pada proses enkripsi dan dekripsi menunjukkan bahwa dalam proses dekripsi suatu data teks berukuran besar algoritma RSA-CRT memiliki efisiensi waktu 50% dibandingkan dengan algoritma RSA standar. Uji coba kecocokan kunci menunjukkan pasangan kunci publik dan kunci privat yang digunakan untuk proses enkripsi dan dekripsi dari RSA maupun RSA-CRT harus berpasangan. Dan hasil pengujian file teks ukuran besar antara 5,10,15 dan 20 mb setelah dienkripsi mengalami kenaikan ukuran file dengan rata-rata 42.757 kb dan setelah didekripsi mengalami pengurangan ukuran file dengan rata-rata 0,007 kb dikarenakan terdapat karakter yang tidak dikenali sistem yaitu simbol bullets list yang mengakibatkan file tidak dapat kembali 100% seperti semula.

UCAPAN TERIMA KASIH

Puji syukur alhamdulillah atas berkat rahmat Allah SWT yang telah memberi segala kemudahan dan kelancaran dalam penyusunan jurnal. Serta semua pihak terkait yang telah senantiasa memberi masukan berupa saran serta semangat yang tiada habisnya sehingga jurnal ini dapat selesai dengan baik dan semoga dapat bermanfaat.

REFERENSI

[1] Menezes, A. J. (1996). *Handbook of Applied Crptography*. CRC Press.

[2] Rivest, R. L. (1978). *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM*, Vol. 21(2): 120-126.

[3] Tilborg, H. C. (2005). *Ecnyclopedia of Cryptography and Security.* New York: SpringerScience+Business Media.

[4] Chenchen Zhang, Yuan Luo, Guangtao Xue. (2016). *A new construction of threshold cryptosystems based on RSA*. Elsevier.

[5] Desi Wulansari, Alamsyah, Fajar Arif Setyawan, Hendi Susanto. (2016). *Mengukur Kecepatan Enkripsi dan Dekripsi Algoritma RSA pada Pengembangan Sistem Informasi Text Security*. Seminar Nasional Ilmu Komputer. Semarang

[6] Arief, Ashari & Saputra, Ragil. (2016). *Implementasi Kriptografi Kunci Publik dengan Algoritma RSA- CRT pada Aplikasi Instant Messaging.* Scientific Journal of Informatics.

[7] Abdulla N. Sura & Al barak Alyaa . (2014). *Text Cryptography Using Chinese Remainder Theorem.* Iraqi Jounal of Science.

[8] Muzakir, Ari & Rahman, Meigi. (2017). *Ujicoba Sistem Keamanan Informasi dengan Algoritma Kriptografi RSA-CRT pada Sistem E-Memo berbasis Mobile*. Simetris.