Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email

Albert Ginting, R. Rizal Isnanto, Ike Pertiwi Windasari
Program Studi Sistem Komputer Fakultas Teknik Universitas Diponegoro
Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia
albertginting@ce.undip.ac.id

Abstract - **In the world of Internet nothing is really safe. There's always a gap in any application made. Likewise in email delivery. To minimize attacks on the data transmission is usually applied cryptography. One fairly popular cryptographic algorithms are RSA algorithm. In this study will discuss the implementation of a cryptographic algorithm RSA encryption and decryption process email. To test created a java-based email client program with message encryption and decryption features messages. This application uses the Java programming language and Netbeans 7.4 as editor. Mail servers used is Google Mail. The initial step of this study was to download email from the Google server and encrypt the message. The second step is decrypt the message to verify whether the message is still the same as the original message before it is encrypted .**

Results from this study is the application that can encrypt and decrypt messages using RSA cryptographic algorithm. With this application is expected to mail delivery is much safer. Because encrypted email will generate a random decimal number of unknown value .

Key Terms: **cryptographic, RSA, encryption, decryption, java**

I. PENDAHULUAN

SEJALAN dengan perkembangan teknologi, semakin mengubah cara masyarakat dalam berkomunikasi. Dulu komunikasi jarak jauh masih menggunakan cara yang konvensional, yaitu dengan cara saling mengirim surat, tetapi sekarang komunikasi jarak jauh dapat dilakukan dengan mudah dan cepat yaitu dengan adanya teknologi seperti *email*, layan pesan singkat (sms), dan Internet yang merupakan salah satu teknologi telekomunikasi yang paling banyak digunakan.

Pada proses pengiriman data (pesan) terdapat beberapa hal yang harus diperhatikan, yaitu : kerahasiaan, integritas data, autentikasi dan non repudiasi. Oleh karenanya dibutuhkan suatu proses penyandian atau pengkodean pesan sebelum dilakukan proses pengiriman. Sehingga pesan yang dikirim terjaga kerahasiaannya dan tidak dapat dengan mudah diubah untuk menjaga integritas pesan tersebut. Ilmu yang mempelajari tentang cara-cara pengamanan data dikenal dengan istilah Kriptografi, sedangkan langkah-langkah dalam kriptografi disebut algoritma kriptografi. Berdasarkan dari kunci yang digunakan algoritma kriptografi dapat dibagi menjadi dua, Algoritma Simetrik dan Algoritma Asimetrik. Dimana Algoritma Simetrik menggunakan satu kunci untuk proses enkripsi dan dekripsinya. Sedangkan Algoritma Asimetrik menggunakan dua kunci berbeda untuk proses enkripsi dan dekripsinya, yaitu kunci umum (*public* key) yang digunakan untuk proses enkripsi yaitu perubahan data teks asli (*plain text)* menjadi teks rahasia (*cipher text*) yang sifatnya tidak rahasia, dan kunci pribadi (*private key*) yang digunakan untuk proses dekripsi yaitu pengembalian data teks rahasia (*cipher text*) menjadi teks asli (*plain text)* yang sifatnya rahasia dan masing-masing pihak memiliki kunci pribadi yang berbeda. Penggunaan kunci pribadi dapat digunakan untuk autentikasi (pengenalan identitas pengirim) dan non repudiasi (pencegahan penyangkalan pengiriman data) karena dalam proses dekripsi dapat diketahui siapa pihak pengirim dengan melihat kunci pribadi yang dipakai. Contoh algoritma kriptografi yang dapat dihandalkan adalah RSA, dimana RSA merupakan proses penyandian kunci asimetrik (*asymmetric key*). Proses perumusan RSA didasarkan pada Teorema Euler, sedemikian sehingga menghasilkan kunci umum dan kunci pribadi yang saling berkaitan. Sehingga meskipun proses enkripsi dan dekripsi menggunakan dua kunci yang berbeda hasilnya akan tetap benar. Kunci umum dan kunci pribadi yang digunakan adalah suatu bilangan prima, dan disarankan bilangan prima yang besar. Hal ini digunakan untuk pencegahan usaha pemecahan teks rahasia, karena semakin besar bilangan prima yang digunakan sebagai kunci maka semakin sulit mencari bilangan besar sebagai faktornya.

Pada Tugas Akhir ini dirancang sebuah sistem purwarupa *mail server* dengan kliennya. *Mail client* dapat melakukan metode enkripsi - dekripsi menggunakan algoritma RSA pada isi pesan bertipe *plaintext*.

Langkah – langkah penelitian yang dilakukan adalah pertama, mengunduh email dari Google *server* kemudian mengenkripsi pesan tersebut. Kedua, pesan yang telah dienkripsi selanjutnya akan didekripsi untuk membuktikan pesan tersebut masih sama dengan pesan asli sebelum dienkripsi dengan menggunakan kunci yang sama.

A. Tujuan

Merancang dan membangun purwarupa *email client* yang mampu melakukan enkripsi dan dekripsi dengan menerapkan ilmu kriptografi RSA sehingga dirasakan aman.

B. Pembatasan Masalah

Tugas akhir ini membatasi batasan masalah yang dibahas untuk menghindari pembahasan diluar penelitian. Batasan masalah pada penelitian ini:

1. Membahas enkripsi dan dekripsi *email* menggunakan algoritma RSA saat mengirim dan menerima *email* bukan membahas keamanan jalur transfernya.
2. Tipe data *email* yang dienkripsi dan dekripsi hanya *plaintext* bukan *attachment file.*
3. *Mail server* menggunakan koneksi protokol SMTP atau IMAP, POP3.

4. Perancangan dan pembuatan *mail client* menggunakan IDE Netbeans, dengan bahasa pemrograman Java.
5. *Mail Server* memakai *server* Google Mail.

Aplikasi yang dibangun hanya sebatas purwarupa sistem tidak menyinggung kepada perancangan tampilan yang atraktif untuk keperluan pengguna.

II. LANDASAN TEORI

A. Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yang terdiri dari kata *kryptos* yang artinya tersembunyi dan *graphia* yang artinya sesuatu yang tertulis sehingga kriptografi dapat juga disebut sebagai sesuatu yang tertulis secara rahasia atau tersembunyi.

Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Kriptografi menurut Rinaldi, Munir (2006) juga didefinisikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek pada keamanan informasi misalnya kerahasiaan, integritas data, otentikasi pengirim/penerima data, dan otentikasi data. Dalam perkembangannya, kriptografi menurut Dony, Ariyus (2008) juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital.

Dalam kriptografi, pesan atau informasi yang dapat di baca disebut sebagai *plaintext* atau *clear text*. Proses yang dilakukan untuk mengubah teks asli (*plaintext)* ke dalam teks rahasia (*chipertext)* disebut enkripsi. Pesan yang tidak terbaca disebut teks rahasia (*chipertext)*. Proses kebalikan dari enkripsi disebut dekripsi. Dekripsi akan mengembalikan teks rahasia (*chipertext)* menjadi teks asli (*plaintext).* Kedua proses enkripsi dan dekripsi membutuhkan penggunaan sejumlah informasi rahasia, yang sering disebut kunci (*key*).

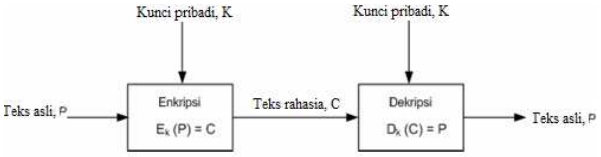
Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

1. Kerahasiaan (*confidentiality*)
   Kerahasiaan adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data
   Integritas adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihakpihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi
   Autentikasi adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu 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 nirpenyangkalan
   Non-repudiasi atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Berbagai macam algoritma kriptografi yang terbagi menjadi dua kelompok dalam hal penggunaan kunci yaitu:

1. Algoritma Sandi Kunci Simetris

Skema algoritma sandi akan disebut kunci-simetris apabila untuk setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kunci yang sama. Pada Gambar 1 dijelaskan bagaimana skema enkripsi dan dekripsi Algoritma Kriptografi Simetris.



Gambar 1. Algoritma Kriptografi Simetris

Beberapa contoh algoritma yang menggunakan kunci-simetris :

1) DES - *Data Encryption Standard*
2) *Blowfish*
3) *Twofish*
4) MARS
5) IDEA
6) 3DES - DES diaplikasikan 3 kali
7) AES - *Advanced Encryption Standard*, yang bernama asli Rijndael

2. Algoritma Sandi Kunci-Asimetris

Skema ini adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Skema ini disebut juga sebagai sistem kriptografi kunci publik karena kunci untuk enkripsi dibuat untuk diketahui oleh umum (*public-key*) atau dapat diketahui siapa saja, tapi untuk proses dekripsinya hanya dapat dilakukan oleh yang berwenang yang memiliki kunci rahasia untuk mendekripsinya, disebut *private-key*. Pada Gambar 2 dijelaskan bagaimana skema enkripsi dan dekripsi Algoritma Kriptografi Asimetris.

Gambar 2 Algoritma Kriptografi Asimetris

B. Kriptografi RSA

Sandi RSA merupakan algoritma kriptografi kunci publik (asimetris). Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Untuk menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Kenyataannya, memfaktorkan bilangan bulat menjadi faktor primanya bukanlah pekerjaan yang mudah. Karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Cara yang bisaa digunakan dalam pemfaktoran adalah dengan menggunakan pohon faktor. Jika semakin besar bilangan yang akan difaktorkan, maka semakin lama waktu yang dibutuhkan. Jadi semakin besar bilangan yang difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA.

Besaran-besaran yang digunakan pada algoritma RSA:

1. p dan q bilangan prima (rahasia)
2. $r = p \cdot q$ (tidak rahasia)
3. $\phi(r) = (p-1)(q-1)$ (rahasia)
4. PK (kunci enkripsi) (tidak rahasia)
5. SK (kunci dekripsi)(rahasia)
6. X (plainteks) (rahasia)
7. Y (cipherteks) (tidak rahasia)

1. ASCII System

Plain teks yang akan dienkripsi dengan RSA Coding merupakan angka-angka, sedangkan pesan yang dikirimkan bisaanya berbentuk teks atau tulisan. Sehingga dibutuhkan suatu kode yang sifatnya universal untuk mengubah pesan teks menjadi plain teks dalam bentuk bilangan. ASCII (American Standard Code for Information Interchange) atau Kode Standar Amerika untuk pertukaran informasi merupakan suatu standar internasional dalam kode huruf dan symbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks.

2. Aritmatika Modulo

Dalam penerapan Teorema Euler pada perumusan algoritma RSA Coding sangat dibutuhkan pemahaman tentang modulo. Modulo sendiri berarti sisa hasil bagi. Misalkan a adalah bilangan bulat dan m adalah bilangan bulat dimana a dan m lebih besar dari 0. Maka operasi $a \bmod m$ (dibaca "a modulo m") memberikan sisa jika a dibagi dengan m. Bilangan m disebut modulus atau modulo, dan hasil modulo m terletak di dalam himpunan $\{0, 1, 2, \ldots, m-1\}$

Contoh :

Diambil a = 20 dan m = 6 . Karena 20 dibagi 6 adalah 3 bersisa 2, maka diperoleh a mod m ≡ 20 mod 6 ≡ 2.

3. Pembangkitan Pasangan Kunci

Sebagai algoritma Asimetris Kriptografi, pengkodean RSA membutuhkan dua kunci yang berbeda untuk enkripsi dan dekripsi. Bilangan yang dipilih sebagai kunci adalah bilangan prima yang besar, dengan alasan pemfaktoran sebuah bilangan hasil perkalian dari dua bilangan prima yang besar menjadi dua bilangan prima yang sesuai akan sangat sulit. Sehingga keamanan dari RSA *Coding* dapat terjamin. Berikut langkah-langkah proses pembangkitan pasangan kunci pada RSA:

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung $r = p \cdot q$. Sebaiknya $p \neq q$, sebab jika $p = q$ maka $r = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari r.
3. Hitung $\phi(r) = (p-1)(q-1)$.
4. Pilih kunci publik, PK, yang relatif prima terhadap $\phi(r)$.
5. Bangkitkan kunci rahasia dengan menggunakan $SK \cdot PK \equiv 1 \pmod{\phi(r)}$.

Perhatikan bahwa $SK \cdot PK \equiv 1 \pmod{\phi(r)}$ ekivalen dengan $SK \cdot PK = 1 + m\phi(r)$, sehingga SK dapat dihitung dengan

$$SK = \frac{1 + m\phi(r)}{PK}$$

4. Proses Enkripsi

Langkah-langkah pada proses enkripsi adalah sebagai berikut :

1. *Plaintext* diubah ke dalam bentuk bilangan. Untuk mengubah *plaintext* yang berupa huruf menjadi bilangan dapat digunakan kode *ASCII* dalam sistem bilangan decimal.
2. *Plaintext* m dinyatakan menjadi blok-blok $x_1, x_2, x_3, \ldots$, sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n-1] , sehingga transformasinya menjadi satu ke satu.
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus $y_i = x_i^{PK} \bmod r$

5. Proses Dekripsi

Langkah-langkah pada proses dekripsi adalah sebagai berikut :

1. Setiap blok *ciphertext* y_i didekripsi kembali menjadi blok x_i dengan rumus $x_i = y_i^{SK} \bmod r$
2. Kemudian blok-blok $m_1, m_2, m_3, \ldots$, diubah kembali ke bentuk huruf dengan melihat kode *ASCII* hasil dekripsi.

6. Kekuatan dan Keamanan RSA

Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini $r = p \times q$. Sekali r berhasil difaktorkan menjadi p dan q, maka $\phi(r) = (p-1)(q-1)$ dapat dihitung. Selanjutnya, karena kunci enkrispi PK diumumkan (tidak rahasia), maka kunci dekripsi SK dapat dihitung dari persamaan $PK \cdot SK \equiv 1 \pmod{\phi(r)}$.

Penemu algoritma RSA menyarankan nilai p dan q panjangnya lebih dari 100 digit. Dengan demikian hasil kali $r = p \times q$ akan berukuran lebih dari 200 digit. Menurut Rivest dan kawan-kawan, uasaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun! (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik).

Untunglah algoritma yang paling mangkus untuk memfaktorkan bilangan yang besar belum ditemukan. Inilah yang membuat algoritma RSA tetap dipakai hingga saat ini. Selagi belum ditemukan algoritma yang mangkus untuk memfaktorkan bilangan bulat menjadi faktor primanya, maka algoritma RSA tetap direkomendasikan untuk menyandikan pesan.

C. Electronic Mail (E-Mail)

Electronic Mail atau dapat disebut dengan E-Mail, merupakan salah satu layanan Internet yang sangat popular dan paling banyak digunakan oleh orang banyak, baik di lingkungan organisasi maupun perusahaan. E-Mail digunakan untuk saling bertukar informasi atau mengirim pesan antara seseorang dengan orang lainnya yang terpisahkan oleh jarak dan kondisi cuaca apapun dengan melewati perangkat telekomunikasi.

1. Sistem Pengiriman Pesan E-Mail

Mail User Agent (MUA) adalah perangkat lunak yang bekerja mengantarkan E-mail kepada user, sedangkan *Message Transport Agent* (MTA). Sebagaimana dengan pengiriman pesan E-Mail kedua aplikasi ini memiliki fungsi, diantaranya yaitu fungsi dari MUA adalah sebagai berikut:

a. Menulis dan membaca pesan E-mail yang masuk baik secara online dan offline karena MUA adalah aplikasi client E-Mail contoh seperti: Microsoft Outlook, Thunderbird, Eudora, Pine dan Pegasus.
b. MUA dapat beroperasi tanpa perlu berkomunikasi.

c. Kemudian user pengguna E-mail client dapat melakukan penyesuaian konfigurasi E-mail sesuai dengan konfigurasi MTA yang ada pada *server* mail.

Sedangkan fungsi dari MTA adalah sebagai berikut:

a. Menentukan header yang digunakan untuk untuk mengenkapsulasi teks E-Mail, termasuk pengiriman dan penerimaan E-mail Address.
b. Menentukan protokol untuk mentransmisikan dan menerima Email.
c. Menentukan protokol bagi client untuk memperoleh E-mail dari *server*.

2. Protokol dan Standar E-Mail

E-Mail memiliki protokol dan standar yang sering dipakai dalam pengiriman dan pembacaan pesan, agar pesan tersebut sampai tujuan. Standar dan Protokol utama yang digunakan dalam layanan E-mail adalah:

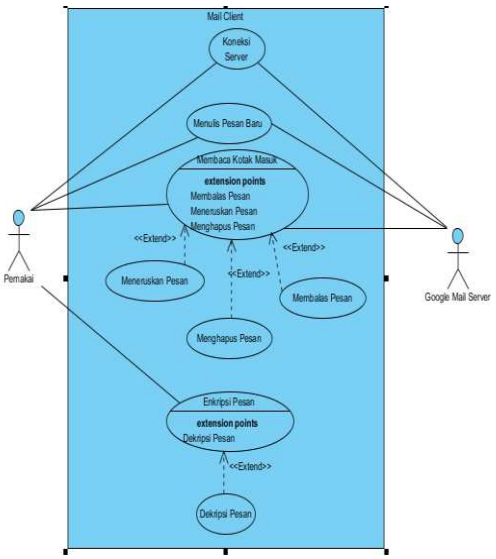
1. RFC 2822 atau Internet Message Format (IMF).
2. RFC 2821 atau Simple Mail Trans*port* Protocol (SMTP).
3. RFC 1939 atau Post Office Protocol versi tiga (POP3).

Selain protokol POP3 yang digunakan dalam medapatkan pesan dari E-mail *server* adapun protokol yang memiliki fungsi sama yaitu Internet Mail Access Protocol (IMAP).

III. PERANCANGAN PENELITIAN

A. Perancangan Diagram Use Case

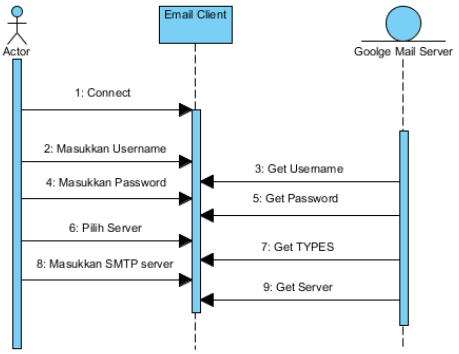
Gambar 3 merupakan diagram *Use Case* yang menggambarkan interaksi antara pengguna dengan sistem yang dirancang beserta fungsionalitas yang diberikan oleh sistem.



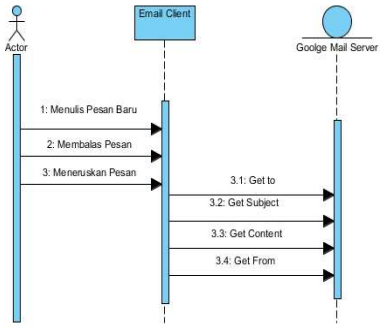
Gambar 3 Diagram *Use Case* sistem

B. Perancangan Diagram Urutan

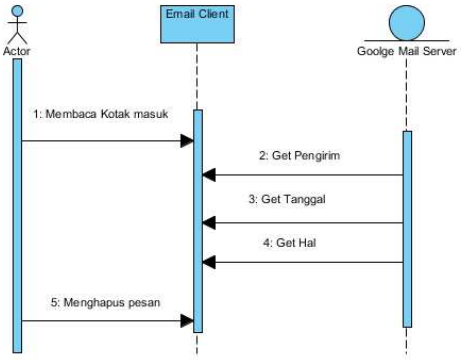
Gambar 4 menunjukkan diagram urutan pengaturan koneksi, Gambar 5 menunjukkan diagram urutan menulis, membalas, dan meneruskan pesan, Gambar 6 menunjukkan diagram uritan membaca dan menghapus pesan, dan Gambar 7 menunjukkan diagram urutan enkripsi dan dekripsi pesan.



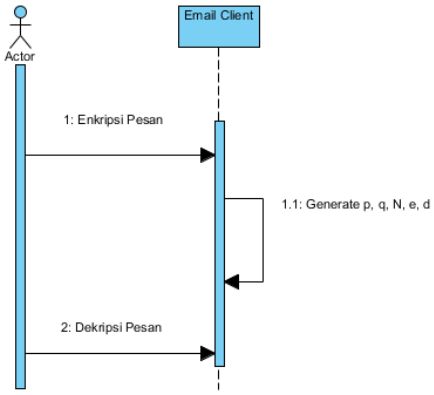
Gambar 4 Diagram urutan pengaturan koneksi



Gambar 5 Diagram urutan menulis, membalas dan meneruskan pesan



Gambar 6 Diagram urutan membaca dan menghapus pesan



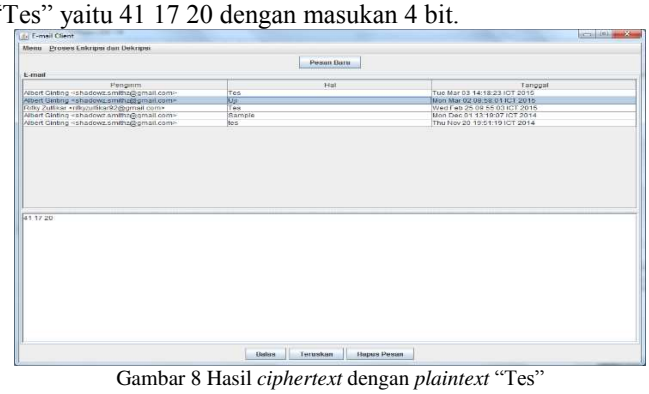
Gambar 7 Diagram urutan enkripsi dan dekripsi pesan

IV. PENGUJIAN SISTEM DAN PEMBAHASAN

Program aplikasi enkripsi dan dekripsi *email* ini dibuat untuk menerapkan ilmu pelajaran kriptografi, yaitu sebuah seni dan bidang keilmuan dalam penyandian informasi atau pesan dengan tujuan menjaga keamanannya.

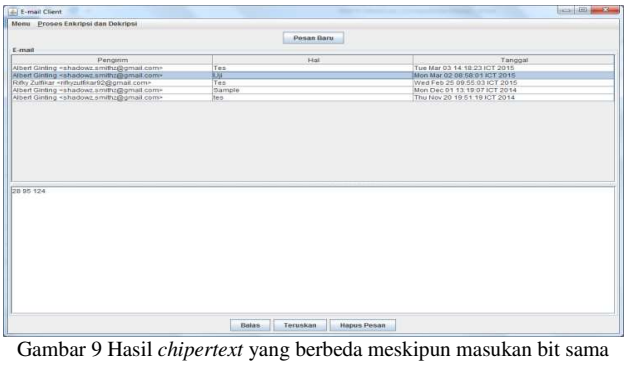
Dalam aplikasi ini telah dibuktikan bahwa pesan dapat dienkripsi dengan menggunakan kunci sehingga pesan yang pada awalnya adalah pesan asli diubah menjadi pesan rahasia. Kemudian, untuk mengembalikan pesan tersebut menjadi pesan asli lagi harus menggunakan kunci yang sama pada saat enkripsi tadi. Proses tersebut dinamakan proses dekripsi.

Pada pengujian kedua dan ketiga, menunjukkan bahwa untuk proses enkripsi maupun dekripsi pesan dengan masukan bit yang berbeda akan menghasilkan *ciphertext* yang berbeda pula, bahkan untuk jumlah bit yang sama bisa menghasilkan *ciphertext* yang berbeda. Hal ini dikarenakan nilai pembangkit kunci yaitu *P* dan *Q* berbeda apa bila masukan bitnya juga berbeda. Gambar 8 menunjukkan hasil enkripsi dari pesan "Tes" yaitu 41 17 20 dengan masukan 4 bit.



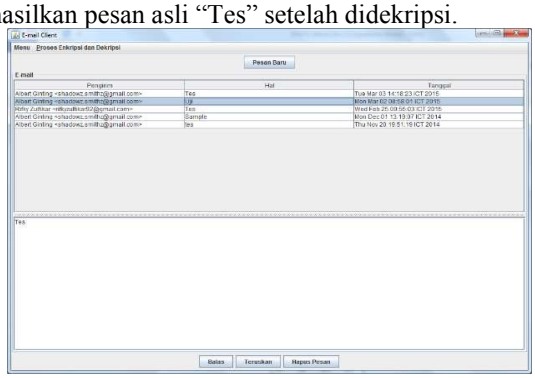
Gambar 8 Hasil *ciphertext* dengan *plaintext* "Tes"

Gambar 9 menunjukkan hasil enkripsi pesan "Tes" dengan masukan 4 bit namun *ciphertext* yang dihasilkan berbeda yaitu 28 95 124.



Gambar 9 Hasil *chipertext* yang berbeda meskipun masukan bit sama

Gambar 10 menunjukkan proses dekripsi menghasilkan pesan asli yang sama meskipun *ciphertext* nya berbeda. *Ciphertext* 41 17 20 dan *ciphertext* 28 95 124 akan menghasilkan pesan asli "Tes" setelah didekripsi.



Gambar 10 Hasil dekripsi c*iphertext* yang berbeda menghasilkan pesan asli yang sama

Perhitungan manual proses enkripsi untuk isi pesan "Tes" Gambar 11 menunjukkan hasil enkripsi dari kata "Tes" oleh aplikasi yang dibuat yaitu 63 17 97

Gambar 11 Hasil enkripsi kata "Tes"

Tahap 1. Ubah teks asli "Tes" menjadi ASCII desimal

Karakter	T	e	s
ASCII (dec)	84	101	115

m dalam desimal = 84101115

Tahap 2. Pilih nilai *P* dan *Q*

Nilai *P* dan *Q* adalah bilangan prima acak yang panjangnya 4 bit (pengujian kedua), nilai $P \neq Q$. Nilai P dan *Q* yang dipakai dalam pengujian kedua $P = 11$, $Q = 13$.

Tahap 3. Tentukan nilai *r, N,* dan *E*

$N = P \,.\, Q$

$N = 11 \,.\, 13$

$N = 143$

$\phi(r) = (p-1)(q-1) = (11\text{-}1)(13\text{-}1)$

$= 120$

Nilai *E* merupakan bilangan relatif prima acak bersifat publik, Faktor Persekutuan Terbesar dari r dan nilainya < r.

$E.gdc(r) = E.gdc(120)$

$= 59$

Tahap 3. Lakukan transformasi satu ke satu untuk *m* (terletak pada rentang 0 – (*n*-1)) hal ini dilakukan agar nilai enkripsi tidak terlampau besar.

Rentang setiap blok $m = 0 - (n\text{-}1) = 0 - 142$

$m = 84101115$

$m_1 = 84$

$m_2 = 101$

$m_3 = 115$

Tahap 4. Enkripsi Pesan

$Y = m^e \bmod N$

$Y_1 = m_1^e \bmod N \equiv 84^{59} \bmod 143 \equiv 63$

$Y_2 = m_2^e \bmod N \equiv 101^{59} \bmod 143 \equiv 17$

$Y_3 = m_3^e \bmod N \equiv 115^{59} \bmod 143 \equiv 97$

Y (*ciphertext*)= 63 17 97, hasilnya sama dengan pengujian kedua.

Tahap 5. Dekripsi Pesan

Hitung nilai *D, D* dapat dicari dengan cara coba-coba, dengan memasukkan nilai *m* satu persatu sampai hasilnya bulat, nilai *D* bersifat rahasia.

$E \,.D \bmod r = 1$

$E.D \equiv 1 \bmod r$

$$D = \frac{1\,(x\,.\,r)}{E} = \frac{1\,(m.\,120)}{59}$$

Dengan mencoba nilai x = 1, 2, 3, 4, ... diperoleh nilai x yang menghasilkan D yang bulat adalah 29. Dan nilai D yang didapat adalah 59.

$$D = \frac{1\ (x\ .\ r)}{E} = \frac{1\ (29.\ 120)}{59}$$

$$= \frac{3840}{59}$$

$$= 59$$

Setelah nilai D didapat langkah selanjutnya ialah mengubah *ciphertext* kembali ke teks awal.

$Y = 63\ 17\ 97 \quad Y_1 = 63 \quad Y_2 = 17 \quad Y_3 = 97$

$m = Y^D \bmod N$

$m_1 = Y_1^D \bmod N \equiv 63^{59} \bmod 143 \equiv 84$

$m_2 = Y_2^D \bmod N \equiv 17^{59} \bmod 143 \equiv 101$

$m_3 = Y_3^D \bmod N \equiv 97^{59} \bmod 143 \equiv 115$

Sehingga, nilai m = 84 101 115

apabila dikonversikan menjadi *string* kembali berdasarkan tabel ASCII maka akan menghasilkan teks asli "Tes".

V. KESIMPULAN DAN SARAN

Pada bagian ini akan dijelaskan kesimpulan dan saran dari hasil penelitian dan pembahasan.

A. *Kesimpulan*

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut.

1. Aplikasi yang menerapkan algoritma kriptografi RSA ini berjalan dengan baik mampu mengirim dan menerima email, dan dapat mengenkripsi dan dekripsi kotak masuk yang diterima.
2. Dengan perangkat lunak ini, tujuan penelitian tercapai yaitu keamanan dalam menerima email terjamin. Ada pengamanan ganda untuk membuka pesan tersandi. Saat mendekripsi pesan yang telah dienkripsi harus memasukkan password terlebih dahulu, apabila masukan password salah pesan tidak akan didekripsi.
3. Perangkat lunak ini hanya mengamankan isi pesan masuk email bukan mengamankan jalur transfer email.
4. Pada aplikasi yang dikembangkan ini, satu pesan asli dapat menghasilkan *ciphertext* yang berbeda-beda, karena proses pembangkitan kunci RSA didasarkan oleh nilai P dan Q yang acak.
5. Pesan kesalahan akan ditampilkan apabila terjadi kesalahan saat memasukkan suatu nilai yang salah saat enkripsi atau dekripsi pesan. Saat enkripsi masukan bit bernai kosong dan saat dekripsi masukan password salah.

B. *Saran*

Untuk pengembangan aplikasi ini di masa yang akan datang, disarankan untuk menambahkan hal-hal sebagai berikut.

1. Perlu dilakukan penelitian lebih lanjut untuk enkripsi dan dekripsi email menggunakan algoritma kriptografi yang lain.
2. Dapat dijadikan sebagai referensi untuk mengembangkan aplikasi enkripsi dan dekripsi email versi *mobile*.

DAFTAR PUSTAKA

[1] Arifin, Rian, and, Oktoviana, Lucky Tri, Implementasi Kriptografi dan Steganografi Menggunakan Algoritma RSA dan Metode LSB, Skripsi S-1, Universitas Negri Malang, 2013.

[2] Ariyus, D., Kriptografi Keamanan Data dan Komunikasi, Graha Ilmu, Yogyakarta, 2008.

[3] D. Chaum, "Untraceable electronic mail, return address and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84– 88, Feb. 1981.

[4] Gomaa, Hassan. *Software Modeling and Design*, Cambridge University Press, New York, 2011.

[5] Kenneth H. Rosen. *Discrete Mathematics and Its Application*. New York:McGraw Hills, 2007.

[6] Muhammad, Arif, *Algoritma RSA sebagai Implementasi Teorema Euler*, http://eprints.undip.ac.id/25130/1/J2A006004-ARIF.pdf, 9 September 2014

[7] Rosyadi, Achmad, *Implementasi Algoritma Kriptografi AES untuk Enkripsi dan Dekripsi Email*, Skripsi S-1, Universitas Diponegoro, Semarang, 2010.

[8] Sasongko, Tri Aditya, Rancang Bangun Email Client pada Perangkat Mobile, http: / / digilib . its . ac . id / public /ITS – Undergraduate -14635 - paperpdf.pdf, 9 September 2014

[9] Saveetha, P, *Study on Improvement in RSA Algorithm and its Implementation*, http://www.advancedsourcecode.com/61-66.pdf, 9 September 2014

[10] Schach, Stephen R., *Classical and Object-Oriented Software Engineering W/ UML and Java,* McGraw-Hill, Inc., New York, 1998

[11] Sulun, Hafni Syaeful, *Penerapan Algoritma Kriptografi RSA dalam Pengiriman Data Melalui Socket Berbasis TCP/IP,* Skripsi S-1, Institut Teknologi Bandung, 2010.

[12] T. Collins, D. Hopkins, and M. Sabin, "Public Key Cryptographic Apparatus and Method,"US Patent #5,848,159. Jan. 1997.

[13] Wiryana, I Made, *Analisis Perbandingan koneksi SMTP dan POP3 Yahoo Mail*, http://idkf.bogor.net/linux-heboh/September%202001/internet-4-elektronik-mail.pdf, 9 September 2014