IMPLEMENTASI KRIPTOGRAFI RSA DENGAN PHP

Muhammad Yasin Simargolang[1]
Program Studi Teknik Informatika, Universitas Asahan,
Jl. Jend. Ahmad Yani Kisaran 21244, Sumatera Utara,
[1]muhammadyasins@gmail.com

***Abstrak* - Kriptografi merupakan seni dan ilmu untuk menjaga keamanan data. Dalam menjaga keamanan data kriptografi mentransformasikan data jelas (plaintext) ke dalam bentuk sandi (ciphertext) yang tidak dapat dikenali. Ciphertext inilah yang kemudian dikirim oleh pengirim kepada penerima. Setelah sampai di penerima, ciphertext tersebut di transformasikan kembali ke dalam bentuk plaintext agar dapat dikenali. Metode yang digunakan adalah metode enkripsi asimetris Rivest Shamir Adleman (RSA). Keamanan algoritme RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci private. Pembuatan aplikasi ini menggunakan bahasa pemrograman PHP. Hasil akhir berupa aplikasi berbasis desktop yang dapat digunakan untuk mengenkripsi file text. Aplikasi ini dapat dijadikan sebagai salah satu cara untuk mengamankan data, karena hanya orang yang berhak yang dapat melihat data yang dienkripsi.**

Kata Kunci : Kriptografi, RSA, Enkripsi – Dekripsi, Text, PHP

***Abstract* - Cryptography is the art and science to maintain data security. In maintaining the security of cryptographic data transforms plaintext data into an unrecognizable ciphertext. Ciphertext is then sent by the sender to the recipient. Upon arriving at the receiver, the ciphertext is transformed back into a plaintext form in order to be recognized. The method used is asymmetric encryption method Rivest Shamir Adleman (RSA). RSA algorithm security lies in the difficulty of factoring large numbers into prime factors. Factoring is done to obtain private key. Making this application using PHP programming language. The end result is a desktop-based application that can be used to encrypt text files. This app can serve as a way to secure data, as only eligible people can view the encrypted data.**

Keywords: Cryptography, RSA, Encryption - Decryption, Text, PHP

I. PENDAHULUAN

A. Latar Belakang

Sangat pentingnya sebuah informasi menyebabkan seringkali informasi diinginkan hanya boleh diakses oleh orang-orang tertentu saja. Jatuhnya informasi kepada pihak lain yang tidak diinginkan (misalnya pihak lawan bisnis) dapat merugikan bagi pihak yang memegang informasi. Untuk itu keamanan dari sistem informasi yang digunakan haruslah terjamin dalam batas yang dapat diterima.

Kerahasiaan harus terdefinisi dengan baik dan prosedur untuk menjaga kerahasiaan informasi harus diterapkan secara berhati-hati. Aspek terpenting dari kerahasiaan adalah pengidentifikasian atau otentikasi terhadap *user*. Identifikasi positif dari setiap *user* sangat penting untuk memastikan efektivitas dari kebijakan yang menentukan siapa yang berhak untuk mengakses data tertentu. Untuk

menjaga keamanan dari *password* atau *username*, biasanya digunakan teknik enkripsi agar kerahasiaan data tersebut terjamin. Jelaslah bahwa teknik enkripsi sangat penting dalam pengamanan data.

Ada beberapa algoritma enkripsi yang biasa digunakan seperti DES, Triple DES, Blowfish, IDEA, RSA *(**R**ivest-**S**hamir-**A**dleman)* dan sebagainya. Khusus untuk RSA merupakan algoritma pertama yang cocok untuk *digital signatur* seperti halnya ekripsi, dan salah satu yang paling maju dalam bidang kriptografi *public key*. RSA masih digunakan secara luas dalam protokol *electronic commerce*, dan dipercaya dalam mengamankan dengan menggunakan kunci yang cukup panjang (Wikipedia Indonesia). Algoritma-algoritma tersebut begitu rumit dan sulit dimengerti dengan dalih "faktor keamanan", katanya semakin sulit algoritma untuk dimengerti, maka semakin aman. Namun bagi para pengguna mereka tidak memikirkan seberapa sulit algoritma dan aplikasinya, yang mereka inginkan hanyalah menjaga kerahasiaan data. Berdasarkan uraian di atas maka penulis mengambil judul **"Implementasi Kriptografi RSA Dengan PHP".**

B. Rumusan Masalah

Berdasarkan uraian latar belakang, maka permasalahan yang akan diteliti adalah:

1. Bagaimana algoritma RSA untuk keamanan data ?
2. Bagaimana analisis kriptografi RSA untuk keamanan data ?
3. Bagaimana rancangan dan implementasi kriptografi RSA ?

C. Tujuan

Penelitian ini bertujuan untuk mengetahui implementasi algoritma RSA untuk keamanan data pada Sistem Informasi Berbasis Web.

D. Manfaat

Manfaat diadakannya penelitian ini adalah sebagai berikut:

1. Bagi peneliti, dapat menerapkan ilmu yang diperoleh di bangku kuliah dalam kehidupan, sehingga ilmu yang dikuasai bukan sekedar teoritis belaka.
2. Bagi kalangan akademik, diharapkan penelitian ini dapat dijadikan perbandingan dengan penelitian serupa dan menjadi bahan pertimbangan untuk penelitian dan pengembangan lebih lanjut.
3. Bagi kalangan umum, diharapkan penelitian ini bermanfaat dan dapat dipertimbangkan untuk dapat dikembangkan.

E. Batasan Masalah

Batasan masalah yang diambil dalam penelitian ini adalah :

1. Implementasi Kriptografi RSA dibangun dengan menggunakan bahasa program PHP
2. Sistem yang dirancang menggunakan UML *(Unified Modelling Language)*
3. Tampilan Sistem berupa tampilan web yang *offline.*

II. LANDASAN TEORI

A. *RSA (Rivest-Shamir-Adleman)*

Penemu pertama algoritma kriptografi kunci asimetri adalah Clifford Cocks, James H. Ellis dan Malcolm Williamson (sekelompok ahli matematika yang bekerja untuk *United Kindom's Government Communication Head Quarters,* agen rahasia Inggris) pada awal tahun 1970. Pada waktu itu temuan itu dipublikasikan dan fakta mengenai temuan tersebut menjadi rahasia hingga tahun 1997.

Algoritma kriptografi kunci asimetri untuk pertama kalinya dipublikasikan pada tahun 1976 oleh Whitfirld Diffie dan Martin Hellman. Dua orang tersebut merupakan ilmuwan dari Stanford University, yang membahas metode pendistribusian kunci rahasia melalui saluran komunikasi umum (*public*), yang kemudian metode tersebut dikenal dengan metode pertukaran kunci Diffie- Hellman (*Diffie-Hellman Key Exchange*).

Ide awal Clifford Cocks ditemukan kembali oleh sekelompok ilmuwan dari Massachussets Institute of Technology pada tahun 1977, sekelompok orang ini adalah Ron Rivest, Adi Shamir, dan Leonard Adleman. Mereka kemudian mempublikasikan temuan mereka pada tahun 1978 dan algoritma kriptografi kunci asimetri yang mereka temukan dikenal dengan nama algoritma kriptografi RSA. RSA itu sendiri merupakan akronim dari nama keluarga mereka, Rivest, Shamir, dan Adleman.

Pada tahun 1983, Massachussets Institute of Technology menerima hak paten atas sebuah makalah berjudul "*Cryptography Communication System and Method*" yang mengaplikasikan penggunaan algoritma kriptografi RSA.

B. *Kriptografi*

1. Definisi Kriptografi

Kriptografi *(cryptography)* berasal dari bahasa Yunani: *"cryptos"* artinya *"secret"* (rahasia), sedangkan *"gráphein"* artinya *"writing"* (tulisan). Jadi kriptografi berarti *"secret writing"* (tulisan rahasia) [10].

Dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Menurut Menezes, kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi. Menurut Schneier, kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan *(message)* [10].

2. Terminologi dalam Kriptografi

2.1 Pesan, *plaintext*, dan *ciphertext*

Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext t*atau teks-jelas *(cleartext).* Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data, dsb) atau yang disimpan di dalam media perekaman (kertas. *storage,* dsb). Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan disandikan ke bentuk lain. Bentuk pesan yang tersandi disebut *ciphertext* atau histogram *(cryptogram). Ciphertext* harus dapat ditransformasi kembali menjadi *plaintext*. Sebagai contoh *plaintext*, uang disimpan di balik buku X, maka *ciphertext*-nya adalah j&kloP#d$gkh*7hA^tr%6^klp..t@.

2.2 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. Entitas dapat berupa orang, mesin, kartu kredit dan sebagainya.

2.3 Enkripsi dan Dekripsi

Proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi *(encryption)* atau *enciphering* (standar nama menurut ISO 7498-2). Proses mengembalikan *ciphertext* menjadi *plaintext*-nya disebut dekripsi *(decryption*) atau *deciphering* (standard nama menurut ISO 7498-2) [10].

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *cipher* [14].

Proses yang dilakukan untuk mengamankan pesan (yang disebut *plainteks*) menjadi pesan yang tersembunyi (disebut *chiperteks*) adalah enkripsi (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext* disebut deskripsi (*decryption*) [12].

2.4 *Cipher* dan Kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enciphering* dan *deciphering,* atau fungsi matematika yang digunakan untuk enkripsi dan deskripsi.

Kriptograf modern mengatasi masalah keamanan algoritma kriptografi dengan penggunaan kunci. Kunci *(key*) adalah parameter yang digunakan untuk transformasi *enciphering* dan *deciphering*. Kunci biasanya berupa *string* atau deretan bilangan.

2.5 *Sistem* Kriptografi

Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Menurut Schneier dalam Munir (2006:7), sistem kriptografi *(cryptographysystem)* adalah kumpulan yang terdiri atas algoritma kriptografi, semua *plaintext* dan *ciphertext* yang mungkin serta kunci.

2.6 Penyadap

Penyadap *(eavesdropper)* adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi sebanyak-banyaknya mengenai sistem kriptografi yang digunakan untuk berkomunikasi dengan maksud untuk memecahkan *ciphertext*. Nama lain penyadap adalah *enemy, adversary, intruder, interceptor, bad guy.*

2.7 Kriptanalisis dan Kriptanalis

Menurut Yusuf Kurniawan [1], kriptanalisis *(cryptanalysis)* adalah ilmu untuk mendapatkan *plaintext* pesan tanpa harus mengetahui kunci secara wajar. Pelakunya disebut kriptanalis. Kriptologi *(cryptology)* adalah studi mengenai kriptografi dan kriptanalisis.

3. Konsep Matematis Algoritma Kriptografi

Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen *plaintext* dan himpunan yang berisi *ciphertext*.

Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut.

Misalkan P menyatakan *plaintext* dan C menyatakan *ciphertext*, maka fungsi enkripsi E memetakan P ke C,

$E(P) = C$

Dan fungsi dekripsi D memetakan C ke P,

$D(C) = P$

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal maka persamaan berikut harus benar,

$D(E(P)) = P$

Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi menjadi,

$EK(P) =$

$C\ DK(C)$

$= P$

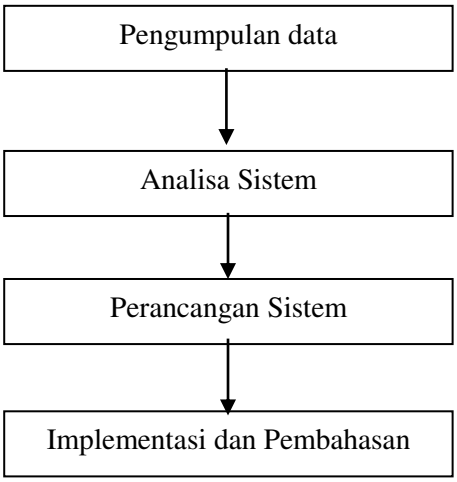
dan kedua fungsi ini memenuhi:

$D_K(E_K(P))$

$= P$

III. METODOLOGI PENELITIAN

A. *Kerangka Kerja Penelitian*



Gambar 3.1 Kerangka Kerja Penelitian

3.1 Pengumpulan Data

Pengumpulan data merupakan hal yang sangat penting dalam sebuah penelitian. Dalam penelitian ini pengumpulan data penulis lakukan melalui :

a. Jurnal

   Jurnal–jurnal yang penulis jadikan sebagai referensi adalah jurnal yang berkaitan dengan sistem kriptografi RSA dengan PHP yang berhubungan dengan judul yang penulis tulis.

b. Buku yang berhubungan dengan penelitian yang dilakukan Buku yang penulis gunakan sebagai referensi adalah buku yang berkaitan dengan judul yang penulis tulis.

3.2 Analisa Sistem

Metode analisa sistem yang digunakan adalah Metode *Deskriptif*. Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan, dianalisa sehingga diperoleh beberapa gambaran yang jelas pada masalah penelitian tersebut.

3.3 Perancangan Sistem

Berikut ini tahapan-tahapan dalam perancangan yang penulis lakukan adalah :

a. Mempelajari Literatur

   Membaca berbagai sumber terkait dengan permasalahan yang akan ditemukan solusinya.

b. Pembuatan Model Sistem

   Membuat pemodelan dalam UML agar lebih mudah dalam tahapan implementasi.

3.4 Implementasi dan Pembahasan

Tahapan ini meng-implementasikan perangkat yang telah dirancang. Apakah sudah bekerja sesuai dengan yang diharapkan. Adapun tahapan peng-implementasian yang dilakukan antara lain :

1. Menguji Sistem kriptografi RSA yang dibuat
2. Melihat hasil enkripsi dari kriptografi RSA.
3. Melihat hasil dekripsi dari kriptografi RSA

IV. ANALISA DAN PEMBAHASAN

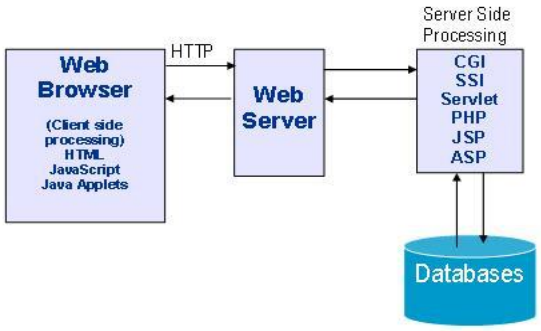
A. *Analisa Sistem*

Pada analisa sistem ini menggunakan metode, yaitu metode *deskriptif* yang telah dijelaskan pada bab III. Di bab ini, penulis hanya membahas pada metode *deskriptif.*

Web *scripting* adalah jenis script yang kita tambahkan (*embedded*) pada halaman web yang sebelumnya hanya disusun dengan sintaks HTML. Penambahan *script* ini mempunyai tujuan tertentu. Misalnya untuk menambahkan informasi jam saat itu, tanggal hari itu, menu yang dinamis (seperti *pull down menu*), kontrol terhadap sebuah *window*, animasi sederhana, maupun untuk validasi form. Salah satu script sisi klien yang sering digunakan adalah Javascript.

Ketika kita membuka sebuah halaman web yang berisi *script* sisi klien, maka secara otomatis *script* tersebut akan ikut dalam halaman web tersebut, baru kemudian *script*nya dieksekusi oleh browser komputer kita. Sehingga proses eksekusi *script* sangat tergantung pada kemampuan *browser* menerjemahkan *script* tersebut.

Manfaat utama dari *script* sisi klien ini adalah bahwa waktu eksekusi relatif lebih cepat dari *script* sisi *server* dan dalam hal tertentu kita lebih memerlukan *client side* daripada *server side* seperti misalnya untuk menu, kontrol elemen halaman web dan validasi form. Tetapi kekurangan atau kelemahan utama dari *client side* ini adalah bahwa *script* yang kita buat pada halaman kita otomatis akan terlihat isinya oleh siapapun yang membuka halaman web tersebut, sehingga dapat saja di *copy paste* untuk digunakan orang lain dengan mudah. Kita juga tentunya bisa dengan mudah mengambil *script JavaScript* dari *website* lain.



Gambar 4.1 Pemrosesan Web sisi *Client* dan *Server*

B. *Perancangan Sistem*

Sebelum masuk ke perancangan sistem, penulis terlebih dahulu memodelkan perancangan yang akan dibuat dengan menggunakan UML (*Unified Modelling Language*).

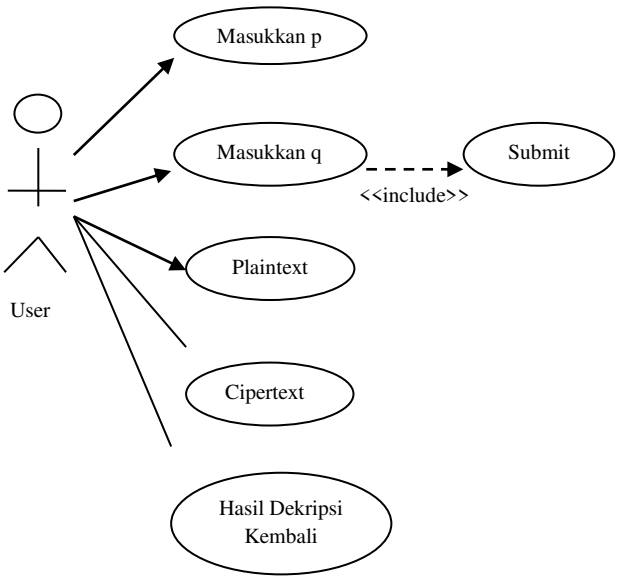
Untuk tahapan perancangan ini ada beberapa tahapan yang penulis lakukan :

1. Penggambaran model rancangan
2. Implementasi sistem kriptografi yang dibuat dengan pemrograman PHP.

1. *Use Case Diagram*

Seperti yang telah dijelaskan sebelumnya *use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Pada diagram ini menekankan "apa" yang diperbuat sistem, dan bukan " bagaimana" membuat sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor

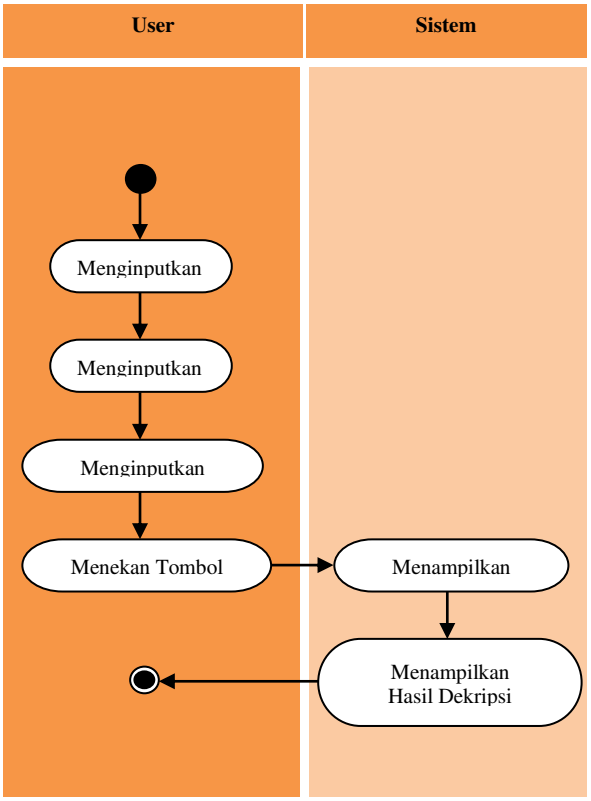
dengan sistem. Gambar 4.1 memodelkan interaksi antara *user* dengan Aplikasi Kriptografi RSA dengan PHP.



Gambar 4.1 *Use Case Diagram* Sistem Kriptografi RSA

2. *Activity Diagram*

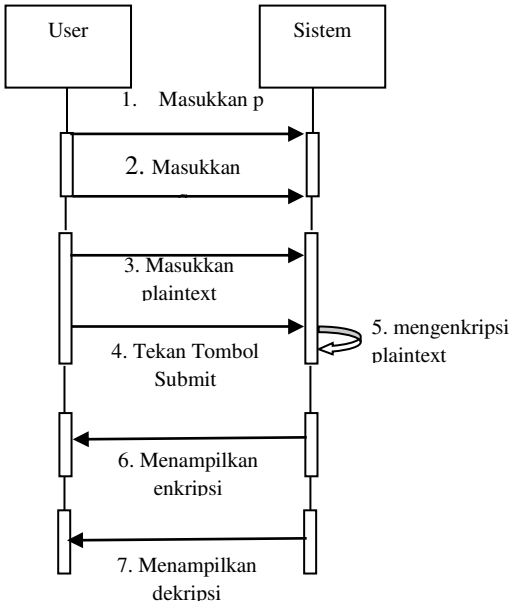
Activity Diagram merupakan suatu penggambaran aliran perilaku pada sistem. *Activity Diagram* ini menunjukkan urutan aktifitas yang dilakukan. Aktifitas adalah eksekusi tugas yang berupa aktifitas fisik atau eksekusi kode. Berikut ini gambar 4.2 *activity diagram* aliran perilaku pada sistem.



Gambar 4.2 *Activity Diagram* Sistem Kriptografi RSA

4. *Sequence Diagram*

Sequence diagram sistem kriptografi RSA dibuat untuk mengenkripsi suatu *plaintext* yang berguna untuk mengunci *plaintext* yang asli, dengan memasukkan kata kunci agar *plaintext* tersebut tidak bisa dilihat oleh pihak lain dan sekaligus mendekripsi suatu enkripsi untuk kembali ke *plaintext* asli. Di bawah ini merupakan gambar *sequence diagram enkripsi.*

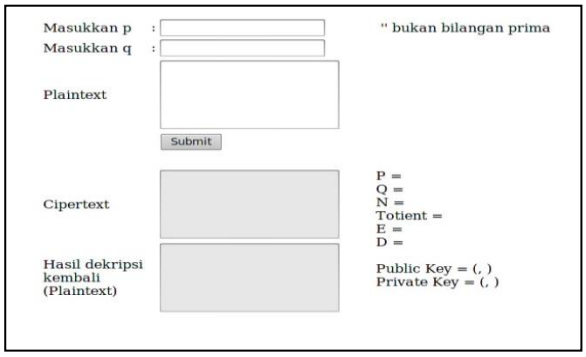


Gambar 4.3 *Sequence Diagram* Sistem Kriptografi RSA

Sequence diagram enkripsi menggambarkanpengguna mengenkripsi *plaintext* dengan cara menekan tombol *submit*, kemudian menampilkan teks enkripsi dan dekripsinya.

C. Rancangan Antar Muka Sistem

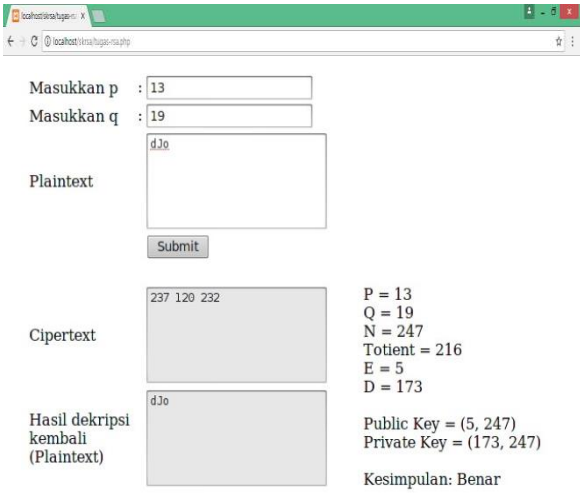
Pada bagian ini, akan direpresentasikan sistem enkripsi dan dekripsi RSA. Berikut ini akan digambarkan langkah-langkah dalam menjalankan program dari awal sampai akhir. Secara keseluruhan perancangan aplikasi ini terdiri dari proses *input* dan *output*. Dalam proses *input* dan *output* terdapat beberapa *form* inputan p dan q, serta *plaintext*. Kemudian *form* tampilan output cipertext dan hasil dekripsi kembali.



Gambar 4.4 Rancangan Antar Muka Sistem Kriptografi RSA

D. Implementasi Sistem

Pada bagian ini, akan diimplementasikan sistem kriptografi RSA. Berikut ini akan digambarkan langkah-langkah dalam menjalankan program dari awal sampai akhir. Secara keseluruhan implementas sistem ini terdiri dari proses *input* dan *output*. Dalam proses *input* dan *output* terdapat beberapa *form* inputan p dan q, serta *plaintext*. Kemudian akan tampil tampilan output *cipertext* dan hasil dekripsi kembali.



Gambar 4.5 Implementasi Sistem

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Dari penelitian ini dapat diberikan hasil kesimpulan dari perancangan sistem kriptografi RSA:

1. Aplikasi enkripsi dan dekripsi ini menggunakan algoritma kriptografi *Rivest Shamir Adleman* (RSA), karena keamanan algoritma RSA ini lebih baik dibanding dengan algoritme *Data Encryption Standard* (DES) yaitu terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima, selain itu juga RSA menggunakan kunci yang berbeda untuk melakukan proses enkripsi dan dekripsi.
2. Dari hasil pengujian respon sistem enkripsi dekripsi dihasilkan waktu akses yang cenderung lebih kecil, apabila pada saat proses enkripsi berlangsung menggunakan bilangan prima dan kunci.
3. Sistem ini dapat membantu data dari rasa kekhawatiran akan pencurian data ketika bertukar data dan dapat menghindari perusakan nama baik akibat manipulasi data.

B. *Saran*

Sebagai saran dari penelitian ini adalah :

1. Sistem ini dapat dikembangkan lagi dengan memberikan inputan *file-file* yang akan dienkripsi dan file-file yang dienkripsi dan dekripsi dapat disimpan dalam satu *folder*.
2. Sistem ini juga dapat dikembangkan lagi untuk dijadikan sistem informasi yang berbasis web.
3. Proses pembangkitan kunci publik dan kunci *private* sebaiknya menggunakan bilangan-bilangan bulat yang besar untuk menghasilkan kunci-kunci yang panjang agar mempersulit percobaan kriptanalis, serta proses *hashing* dapat menggunakan fungsi-fungsi *hash* lainnya selain MD5 yang memilki kemungkinan kolisi atau sinonim yang kecil.

DAFTAR PUSTAKA

[1] Andri, Yuli M. 2009. *Implementasi Algoritma DES,RSA, dan Algoritma Kompresi LZW Pada Berkas Digital.* Skripsi. USU. Dipublikasikan.

[2] Arazi, T. (n.d) *Penggunaan Teori Bilangan pada Algoritma RSA, Protokol Diffie- Hellman, dan Pencegahan Terhadap Timing Attacks Online*. Available at http://www.mail.informatika.org/renaldi/Matdis/2006-2007/Makalah/Makalah0607-05.pdf. [accessed 19/03/09].

[3] Azis, M. F. 2005. *Object Oriented Programming dengan PHP 5*. Jakarta : Elex Media Komputindo.

[4] *Basis Data*. http://id.wikipedia.org/wiki/Basis_data

[5] Fathansyah. 2004. *Buku Teks Komputer Sistem Basis Data*. Bandung : Informatika.

[6] Hariyanto, B. 2004. *Sistem Manajemen Basis Data. Bandung* : Informatika. Kadir, A. 1999.

[7] *Konsep dan Tuntunan Praktis Basis Data*. Yogyakarta : Andi. Kadir, A. 1999. *Penuntun Praktis Belajar SQL*. Yogyakarta : Andi.

[8] Kadir, A. 1999. *Tuntunan Praktis Belajar Database Menggunakan MySQL*. Yogyakarta : Andi.

[9] Munir, Rinaldi. 2005. *Matematika Diskrit*. Bandung : Informatika.

[10] Munir, Rinaldi. 2006. *Kriptografi. Bandung* : Informatika.

[11] Nugroho, Bunafit. 2004. *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*. Yogyakarta : Gava Media.

[12] Rahardjo, B. 1998. *Keamanan Sistem Informasi Berbasis Internet*. Bandung : PT. Insan Ifonesia.

[13] Riyanto, Djalal. 2004. *Buku Ajar Basis Data*. Semarang : Universitas Diponegoro.
RSA. http://id.wikipedia.org/wiki/RSA

[14] Wahana Komputer. 2003. *Memahami Model Enkripsi dan Security Data*. Yogyakarta: Andi Offset.