Penerapan Kriptografi RSA Dalam Mengamankan File Teks Berbasis PHP

Dicky Apdilah[1], Heru Swanda[2]
[1]*Amik Intelcom Global Indo*
Kisaran, Indonesia
[2]*Program Studi Teknik Informatika, Fakultas Teknik, Universitas Asahan*
Jln. Jend. Ahmad Yani, Kisaran, Indonesia
dickyapdilah14@gmail.com[1], heruswanda@gmail.com[2]

***Abstrak* - Seiring dengan perkembangan teknologi komunikasi kebutuhan manusia dalam penggunaan teknologi semakin meningkat, terlebih dalam penyimpanan data. Salah satu cara untuk meningkatkan keamanan terhadap data yaitu dengan menggunakan metode kriptografi. Algoritma RSA *(Rivest Shamir Adleman)* merupakan salah satu metode dalam cabang ilmu kriptografi, dimana RSA adalah jenis kriptografi asimetris yang menggunakan 2 kunci, yaitu kunci publik dan private. Masalah untuk meningkatkan keamanan kunci publik dan kunci privat pada RSA (*Rivest Shamir Adlema*) maka diperlukan metode *Linear Congruential Generator (LCG)*, LCG digunakan untuk menghasilkan sekumpulan bilangan acak sampai ke-n, dimana sekumpulan bilangan acak tersebut akan di ambil yang memiliki nilai bilangan prima. Salah satu metode untuk membangkitkan bilangan prima adalah algoritma *The Sieve Of Eratosthenes*, algoritma *The Sieve Of Eratosthenes* merupakan sebuah algoritma klasik untuk menentukan seluruh bilangan prima sampai bilangan ke-n yang di tentukan. Cara kerja *metode The Sieve Of Eratosthenes* adalah dengan melakukan eliminasi bilangan yang bukan bilangan prima, sehingga menghasilkan kumpulan bilangan prima. Bilangan prima yang telah di bangkitkan dengan algoritma *The Sieve Of Eratosthenes* nantinya akan digunakan untuk kunci publik dan kunci private pada kritografi RSA.**
Kata Kunci - RSA, LCG,The Sieve of Eratosthenes.

I. PENDAHULUAN

Salah satu cara untuk meningkatkan keamanan terhadap data atau file yaitu dengan menggunakan metode kriptografi. Pada kriptografi modern, algoritma yang digunakan tidak dirahasiakan sebab setiap kali algoritma diketahui pihak lain, maka kriptografer harus membuat algoritma baru, dengan demikian cukup kuncinya yang harus dirahasiakan dan benar-benar dijaga keamanannya. Berdasarkan jenis kunci yang digunakan, kriptografi terbagi atas dua metode, yaitu kriptografi kunci simetris dan kriptografi kunci asimetris.

Algoritma RSA *(Rivest Shamir Adleman)* merupakan salah satu metode dalam cabang ilmu kriptografi, dimana RSA adalah jenis kriptografi asimetris yang menggunakan 2 kunci, yaitu kunci publik dan private. Algoritma kriptografi RSA didesain sesuai fungsinya sehingga kunci yang digunakan untuk enkripsi berbeda dari kunci yang digunakan untuk dekripsi. Kunci untuk enkripsi pesan disebut publik, sedangkan kunci untuk mendekripsi pesan yang diterima disebut private.

Untuk mempermudah dalam menentukan bilangan prima yang akan digunakan sebagai kunci publik dan private serta meningkatkan keamanan Kriptografi RSA *(Rivest Shamir Adleman)* dapat dilakukan dengan algoritma *Linear Congruential Generator (LCG)* dan algoritma *The Sieve Of Eratosthenes* , *LCG* dapat diterapkan untuk menghasilkan sekumpulan nilai acak sampai bilangan ke-n. Sekumpulan bilangan acak yang telah di bangkitkan menggunakan LCG akan diambil yang memiliki sifat bilangan prima. Salah satu metode untuk membangkitkan bilangan prima tersebut adalah algoritma *The Sieve Of Eratosthenes*, Cara kerja metode *The Sieve Of Eratosthenes* adalah dengan melakukan eliminasi bilangan yang bukan bilangan prima, sehingga menghasilkan kumpulan bilangan prima.

Berdasarkan Penelitihan jurnal sebelumnya, "Proses penentuan bilangan prima digunakan fungsi *Math.sqrt* yang sesuai dengan metode *The Sieve Of Eratosthenes* untuk menentukan nilai akar kuadrat dari suatu bilangan sehingga diharapkan pengecekan bilangan prima bisa dilakukan lebih cepat dan mengemat memori" (Safri, 2013).

Bilangan prima yang telah di bangkitkan nantinya digunakan sebagai kunci publik dan kunci private pada RSA.

II. LANDASAN TEORI

A. *Kriptografi*

Kata Kriptografi berasal dari bahasa Yunani,"*Kriptos"* yang berarti tersembunyi atau rahasia dan "*graphien"* yang berarti menulis. Dari sini dapat diartikan bahwa kriptografi adalah suatu praktek atau cara untuk mendapatkan komunikasi yang aman walau ada pihak ketiga yang berusaha mencuri dengar (William, 2011).

B. *Sejarah Kriptografi*

Menurut Munir (2012:205), kriptografi sudah lama digunakan oleh tentara *Sparta* di yunani pada permulaan tahun 400 SM. Mereka menggunakan alat yang disebut *scytale.* Alat ini terdiri dari sebuah pita panjang dari daun papyrus yang dililitkan pada sebatang tabung silinder. Pesan yang dikirim ditulis horizontal (baris per baris). Bila pita itu dilepaskan. Maka huruf huruf didalamnya telah tersusun membentuk pesan rahasia. Untuk membaca pesan, penerima melilitkan kembali silinder yang diameternya sama dengan diameter silinder pengirim.

Gambar 1 Scytale yang digunakan oleh tentara Yunani untuk transposisi Pesan
Sumber : (Munir. 2012:205)

C. Algoritma Kriptografi RSA

Dari sekian banyak algoritma kunci publik yang pernah dibuat, algoritma yang paling populer adalah algoritma *RSA*. Algoritma *RSA* dibuat oleh 3 orang peneliti dari MIT (*Massachusetts Institute of Technology*) pada tahun 1976, yaitu: *Ron* ***(R)****ivest, Adi* ***(S)****hamir*, dan *Leonard* ***(A)****dleman*. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmatika modulo. Baik kunci enkripsi maupun kunci dekripsi keduanya berupa bilangan acak. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma *RSA* tetap terjamin (Sadikin, 2012 : 249).

Para pembuat RSA ini melihat bahwa operasi *m*k mod *n* menghasilkan nilai yang relative acak hubungan terhadap *m*. Berikut ini akan disampaikan pembentukan kunci privat dan kunci publik dengan RSA :

1. Pilih dua bilangan prima *p* dan *q* secara acak. Bilangan ini harus cukup besar yaitu maksimal 3 digit.
2. Hitung $n = pq$. Untuk kemudian bilangan *n* disebut parameter sekuriti.
3. Pilih bilangan *e* secara acak di mana *e* tidak memiliki faktor pembagi yang sama dengan (p-1)(q-1) selain bilangan 1. Atau dengan kata lain bersifat relatif prima.
4. Hitung *d* sedemikian sehingga $ed = 1 \pmod{\Phi(n)}$. Dengan menggunakan sebuah algoritma yang disebut algoritma Euclid akan menghitung *d* sehingga, $d = (1 + k\,\Phi(n))/e$
5. Bilangan *n* dan *e* kita sebarkan ke publik. *e* ini adalah yang akan menjadi kunci publik. *d* menjadi kunci privat. Sementara itu bilangan *p* dan *q* dihilangkan, dan dicegah agar tidak pernah sampai bocor ke publik.

Kini sudah didapatkan sebuah kunci publik dan kunci privat. Selanjutnya berikut ini adalah algoritma untuk menyandi dan menterjemahkan pesan :

1. Untuk menyandi sebuah pesan P dengan menggunakan kunci publik *e*, kita melakukan operasi $P^e \bmod n$, sementara untuk membuka pesan tersandi C dengan menggunakan kunci privat, kita lakukan $C^d \bmod n$
2. Untuk memudahkan enkripsi dan dekripsi maka pesan dibagi menjadi beberapa blok yang kecil.

Algoritma di atas adalah algoritma yang digunakan dalam penyandian RSA,
maka hanya menggunakan operasi pemangkatan bilangan dan modulus bilangan, dalam melakukan proses enkripsi dan dekripsi sebuah pesan. Kesederhanaan inilah yang menjadikan RSA menjadi popular karena relatif mudah dimengerti.
Adapun dari algoritma di atas dapat disederhanakan seperti:
Kunci Publik : *e* = relative prima (p-1)(q-1)
Kunci Privat : $d = (1 + k\,\Phi(n))/e$
Enkripsi : $C = P^e \bmod n$
Dekripsi : $P = C^d \bmod n$

D. Pembangkit Bilangan Acak Linear Congruential Generator (LCG)

Bilangan acak adalah bilangan yang tidak dapat diprediksi kemunculannya. Tidak ada komputasi yang benar – benar menghasilkan deret bilangan acak secara sempurna. Banyak algoritma atau metode yang dapat digunakan untuk membangkitkan bilangan acak salah satunya adalah pembangkit bilangan acak *Linear Congruential Generator* (*LCG*). *LCG* adalah salah satu pembangkit bilangan acak tertua dan sangat terkenal. *LCG* didefinisikan dalam relasi rekurens yaitu Schneier, (1996) : $xn = (ax_{n-1} + b) \bmod m$ (Munir, 2012 : 217)**.**
Yang dalam hal ini:
Xn = bilangan acak ke-n dari deretnya
x_{n-1} = bilangan acak sebelumnya
a = faktor pengali
b = penambah (*increment*)
m = modulus (*a*, *b* , dan *m* semuanya konstanta)
Misalkan"
a = 9 b = 6 X_0= 45 m = 100
Penyelesaian :
$Xn = (ax_{n-1} + b) \bmod m$
$X_1 = (9*X_{1-1} + 6) \bmod 100$
$= (9*X_0 + 6) \bmod 100$
$= (9*45 + 6 \bmod 100$
$= 411 \bmod 100$
$= 11$
$X_2 = (9*X_{2-1} + 6) \bmod 100$
$= (9*X_1 + 6) \bmod 100$
$= (9*11 + 6) \bmod 100$
$= 105 \bmod 100$
$= 5$
$X_3 = (9*X_{3-1} + 6) \bmod 100$
$= (9*X_2 + 6) \bmod 100$
$= (9*5 + 6) \bmod 100$
$= 51 \bmod 100 = 51$
$X_4 = (9*X_{4-1} + 6) \bmod 100$
$= (9*X_3 + 6) \bmod 100$
$= (9*51 + 6) \bmod 100$
$= 465 \bmod 100 = 65$

$X_5 = (9*X_{5-1} + 6) \bmod 100$
$= (9*X_4 + 6) \bmod 100$
$= (9*65 + 6) \bmod 100$
$= 591 \bmod 100$
$= 91$

Maka rangkaian bilangan acak yang dibangkitkan adalah :

TABEL 1
Contoh Bilangan Acak Yang Dihasilkan Dari Persamaan *LCG*

N	Xn
0	11
1	5
2	51
3	65
4	91
5	25
6	31
7	85
8	71
9	45
10	11
11	5
12	51
13	65
14	91

E. Algoritma Pembangkit Bilangan Prima The Sieve of Eratosthenes

Ada berbagai metode yang dapat digunakan untuk menghasilkan sebuah bilangan prima.Untuk menghasilkan bilangan prima yang besar dengan menggunakan ruang memori dan waktu. Secara umum pembangkitan bilangan prima dapat dikelompokkan menjadi dua, yaitu dengan membangkitkan bilangan prima dari bilangan prima terkecil dengan pengujian yang menghasilkan 100% bilangan prima atau dengan membangkitkan bilangan acak dan menguji kemungkinan bilangan tersebut prima (Haro, 2011).

Sieve of Eratosthenes sebuah algoritma klasik untuk menemukan seluruh bilangan prima sampai ke sebuah n yang ditentukan. Mulai dengan *array of integer* yang belum dicoret dari 2 ke n. Integer pertama yang belum dicoret yaitu 2,adalah bilangan prima pertama. Coret seluruh kelipatan dari bilangan prima ini. Ulangi pada integer selanjutnya yang belum dicoret.

Sebagai contoh, berikut adalah *array* pada awalnya: 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 2526 27.

Karena 2 belum dicoret, maka 2 adalah bilangan pertama.Coret seluruh kelipatan 2, yaitu 4, 6, 8, 10, 12,dst. **2** 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25 ~~26~~ 27.

Integer selanjutnya yang belum dicoret adalah 3, maka 3 adalah prima dan coret seluruh kelipatan 3, seperti 6, 9. 12,dst. **2 3** ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24~~ ~~25 26 27~~ .

5 adalah bilangan prima selanjutnya dan coret seluruh kelipatan 5. Satu-satunya bilangan yang dicoret dalam *range* ini adalah 25. **2 3** ~~4~~ **5** ~~6~~ **7** ~~8 9 10~~ **11** ~~12~~ **13** ~~14 15 16~~ **17** ~~18~~ **19** ~~20 21 22~~ **23** ~~24 25 26 27~~ . Maka bilangan prima setelah 2,3 dan 5 yaitu7, 11, 13, 17, 19, dan 23.

F. Bahasa Pemograman PHP

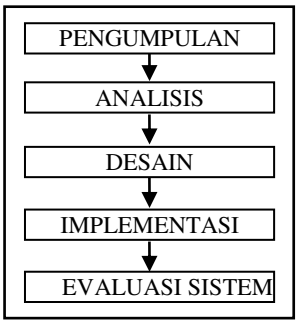
PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis. PHP sering juga digunakan untuk membangun sebuah CMS (Madcoms,2016:2).

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Disebut bahasa pemrograman Server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client –side seperti Java Script yang diproses pada web browser (client).

III. METODE PENELITIAN

A. Kerangka Kerja

Kerangka kerja yang dilakukan dalam penyusunan penelitian ini adalah mempelajari materi yang berkaitan dengan Aplikasi Pengamanan File Teks dengan menggunakan Kriptografi RSA berbasis PHP sebagai konsep penyelesaian masalah, adapun kerangka kerja penelitian ini dapat dilihat seperti gambar dibawah ini :



Gambar 2 Kerangka Kerja

B. Uraian Kerangka Kerja

Berdasarkan kerangka kerja pada gambar 3.1 maka dapat diuraikan langkah-langkah kerja sistem perancangan aplikasi sebagai berikut :

1. Pengumpulan Data
   Yaitu pengumpulan data dan informasi dengan cara mencari artikel, jurnal, dan *e-book* di Internet yang dapat dijadikan acuan pembahasan yang berhubungan dengan judul penelitian ini.
2. Analisis
   Analisis Identifikasi masalah. Permasalahan diidentifikasi sebagai suatu hal yang menghambat tujuan penelitian. Permasalahan harus ditindaklanjuti untuk ditemukan pemecahannya. Analisis

Kebutuhan sistem yaitu mengidentifikasi kebutuhan-kebutuhan yang diperlukan dalam pembuatan aplikasi pengamanan file teks dengan menggunakan kriptografi RSA berbasis PHP baik kebutuhan masukan *(Input)* ataupun kebutuhan keluaran *(Output),* kebutuhan perangkat keras *(Hardware)* dan kebutuhan perangkat lunak *(Software)* dalam pembuatan aplikasi dan penggunaan aplikasi.

3. Desain
   Pada tampilan desain yang dirancang merupakan dari tampilan pembangkit kunci, tampilan enkripsi, tampilan dekripsi.
4. Implementasi
   Yaitu sebuah metode penelitian berupa suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Implementasi biasanya dilakukan setelah perencanaaan sudah dianggap bagus dan layak untuk diluncurkan. Penelitian terhadap implementasi pada akhirnya sangat menentukan bagus atau tidaknya sistem yang dirancang dan setelah itu akan diadakan proses evaluasi dengan tujuan peningkatan kualitas sistem yang dirancang.
5. Evaluasi sistem
   Evaluasi sistem pada perancangan aplikasi ini merupakan proses pengkodean yang selesai, maka akan dilakukan proses pengujian terhadap program yang dikerjakan untuk mengetahui apakah program sudah berjalan dengan benar dan apabila ada kesalahan pada aplikasi maka akan dilakukan perbaikan sesuai dengan perancangan aplikasi yang dilakukan

C. Metode Penelitian

Metode dalam perancangan aplikasi penelitian ini, peneliti menggunakan metode studi pustaka berupa pengumpulan data dari artikel, jurnal, dan *e-book*, yang nantinya digunakan untuk memperoleh data-data yang dibutuhkan dalam pembuatan Aplikasi Pengamanan File, sehingga yang nantinya aplikasi yang digunakan benar-benar bermanfaat baik bagi pengguna sendiri maupun orang lain.

D. Metodologi Penelitian

Adapun metodologi penelitian yang digunakan peneliti untuk menyelesaikan permasalah yang diangkat yaitu :

1. Studi Literatur
   Pengumpulan data yang berkaitan dengan permasalahan dengan cara membaca-baca buku, berupa jurnal ataupun artikel dan membaca bahan-bahan sumber lainnya baik dari perpustakaan maupun luar perpustakaan.

2. Analisis
   Peneliti menganalisa permasalahan terhadap objek yang telah dipilih, baik itu menganalisa sistem berupa kebutuhan masukan (input), kebutuhan keluaran (output), kebutuhan perangkat keras (hardware), dan kebutuhan perangkat lunak (software) dalam pembuatan aplikasi.
3. Desain Sistem
   Pada tahap ini peneliti melakukan perancangan program, membuat desain aplikasi pengamanan file dengan menggunakan kriptografi RSA berbasis PHP.
4. Uji Coba
   Melakukan pengujian program, menangani dan memperbaiki kesalahan yang ada pada aplikasi pengamanan file dengan menggunakan kriptografi RSA berbasis PHP untuk pemrosesan data pakar agar dapat berjalan dengan baik sesuasi dengan yang diinginkan.

E. Obyek Penelitian

Obyek penelitan ini terdiri atas bahan dan peralatan penelitian.

F. Bahan Penelitian

Bahan penelitian diambil dari sumber data – data yang diperoleh dari buku literatur, studi pustaka mengenai Kriptografi Untuk Keamanan Jaringan, Matematika Diskrit, Dreamweaver CS6, Logika dan Algoritma, Pemrograman PHP.

G. Peralatan Penelitian

Peralatan yang digunakan terdiri dari perangkat keras (*hardware*) yaitu sebuah komputer dan perangkat lunak (*software*) dengan konfigurasi minimal sebagai berikut :

1. Spesifikasi Hardware
   a. Laptop HP 240 G5
   b. Processor : Intel Core i3
   c. Monitor : 14”
   d. RAM : 4,00 GB
   e. Hardisk : 500 GB
2. Aplikasi yang digunakan
   a. Windows 7
   b. Microsoft Office 2010
   c. Adobe Dreamweaver CS6
   d. XAMPP

H. Defenisi Operasional Variabel

Definisi operasional ini dimaksudkan untuk menghindari kesalahan pemahaman dan perbedaan penafsiran yang berkaitan dengan istilah – istilah dalam judul penelitian ini. Definisi operasional variabel menjelaskan tipe – tipe variabel yang dapat diklasifikasikan berdasarkan fungsi variabel dalam hubungan antar variabel serta skala pengukuran yang digunakan.

Variabel dalam penelitian ini terdiri dari variabel bebas (X) dan variabel terikat (Y). Variabel bebas merupakan faktor stimulus atau input yaitu faktor yang dipilih oleh peneliti untuk melihat pengaruh terhadap gejala yang diamati. Variabel terikat yaitu faktor yang diamati dan di ukur untuk mengetahui efek variabel bebas. Berdasarkan rumusan masalah yang telah dibuat, maka dirumuskan variabel – variabel penelitian sebagai berikut :

1. Variabel bebas (X)
   Variabel bebas dalam penelitian ini yaitu tentang masukan pengamanan file.
2. Variabel terikat (Y)
   Variabel terikat dalam penelitian ini yaitu tampilan hasil dari pengamanan file yang telah dilakukan dengan jelas sehingga *user* mendapatkan informasi-informasi yang dibutuhkan dan *user* dapat mengerti serta memahami penjelasan tersebut dengan baik.

I Teknik Pengumpulan Data

Dalam hal ini metode penelitian yang digunakan adalah metode dengan cara mengumpulkan dan menggambarkan data mengenai keadaan secara langsung dari lapangan atau tepatnya yang menjadi objek penelitian untuk mendapatkan data secara relevan. Teknik pengumpulan data yang dilakukan dalam mencari dan mengumpulkan data serta mengolah informasi yang diperlukan menggunakan beberapa metode sebagai berikut :

1. Penelitian (observasi)
   Merupakan cara pengumpulan data dimana peneliti tidak memiliki kendali sama sekali terhadap pemunculan respon objek yang diamati, kecuali dalam menentukan faktor yang diamati dan memeriksa ketelitian data.
2. Studi Pustaka
   Yaitu memperoleh data dengan cara membaca dan mempelajari buku-buku dan *literature-literature* yang berhubungan dengan teori dan laporan penelitian ini

J. Analisis Data

Adapun analisis data yang dilakukan peneliti adalah sebagai berikut :

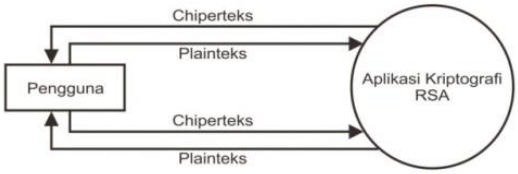
1. Data diseleksi dan dikelompokan dengan kebutuhan untuk menjawab masalah penelitian.
2. Data diolah sesuai dengan masalah penelitian.

Analisa data dengan menggunakan kata-kata yang sederhana sebagai jawaban terhadap masalah.

IV. ANALISIS DAN IMPLEMENTASI

A. Diagram Konteks

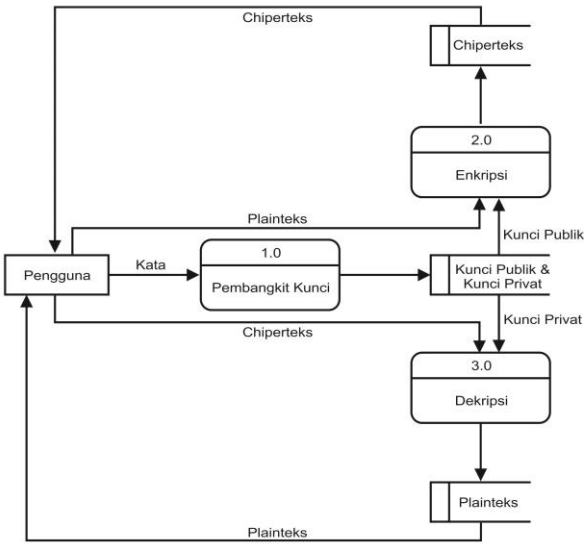
Diagram konteks adalah suatu diagram alir tingkat tinggi yang menggambarkan seluruh jaringan, masukan dan keluaran. Berikut adalah diagram konteks pada aplikasi kriptografi RSA dalam mengamankan file berformat Txt berbasis PHP.



Gambar 3 Diagram Konteks Aplikasi Kriptografi RSA

B. Data Flow Diagram

Rancangan *Data Flow Diagram* (DFD) digunakan untuk menggambarkan suatu sistem yang sedang berjalan. Berikut adalah *Data Flow Diagram* (DFD) pada aplikasi kriptografi RSA dalam mengamankan file berformat Txt berbasis PHP.



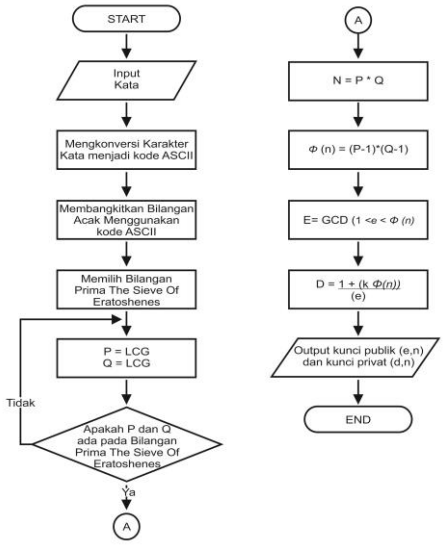
Gambar 4 *Data Flow Diagram Aplikasi Kriptografi RSA*

C. Rancangan Flowchart Pembangkitan Kunci

Algoritma *RSA* melibatkan kunci publik dan kunci privat Kunci publik dapat diketahui semua orang dan digunakan untuk mengenkripsi pesan. Pesan dienkripsi dengan kunci publik hanya dapat didekripsi menggunakan kunci privat. Kunci untuk algoritma *RSA* yang dihasilkan dengan cara berikut:

1. Pilih dua bilangan prima p dan q secara acak. Bilangan ini harus cukup besar yaitu maksimal 3 digit.
2. Hitung $n = pq$. Untuk kemudian bilangan n disebut parameter sekuriti.
3. Pilih bilangan e secara acak di mana e tidak memiliki faktor pembagi yang sama dengan (p-1)(q-1) selain bilangan 1. Atau dengan kata lain bersifat relatif prima.
4. Hitung d sedemikian sehingga $ed = 1 \pmod{\Phi(n)}$. Dengan menggunakan sebuah algoritma yang disebut algoritma Euclid akan menghitung d sehingga, $d = (1 + k\,\Phi(n))/e$
5. Bilangan n dan e kita sebarkan ke publik. e ini adalah yang akan menjadi kunci publik. d menjadi kunci privat. Sementara itu bilangan p dan q dihilangkan, dan dicegah agar tidak pernah sampai bocor ke publik.

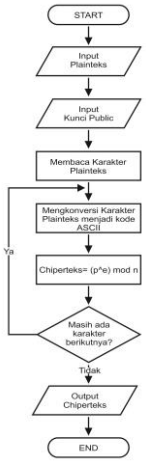
Kunci publik terdiri dari modulus n dan e (atau enkripsi) eksponen publik. Kunci privat terdiri dari modulus n dan (atau dekripsi) eksponen d yang harus dirahasiakan. Untuk perancangan aplikasinya dapat dilihat pada flowchart berikut :



Gambar 5 Flowchart Pembangkit Kunci RSA

D. Rancangan Flowchart Enkripsi

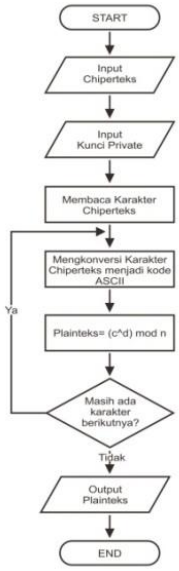
Dalam proses enkripsi data penyandian data menggunakan kunci publik *(n, e)* awalnya data yang diinput akan diubah kedalam bentuk *ASCII* pada setiap karakternya, selanjutnya plainteks akan dienkripsikan menjadi *chiperteks* dengan menggunakan metode *exponentiation* dengan mengkuadratkan yaitu $C = p^{e} \pmod{n}$ Perancangan aplikasinya dapat dilihat pada *flowchart* berikut :



Gambar 6 *Flowchart* Proses Enkripsi

E. Rancangan Flowchart Dekripsi

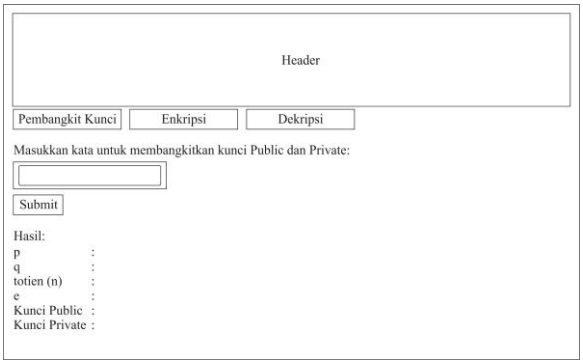
Dalam proses dekripsi mengubah data yang sudah disandikan (*chiperteks*) menggunakan kunci publik *(e, n)* menjadi pesan awal (*plainteks*) dapat dilakukan dengan menggunakan kunci privat *(d, n),* chiperteks yang diinput akan diubah kedalam bentuk *ASCII* pada setiap karakternya, selanjutnya *chiperteks* akan didekripsikan menjadi *plainteks* dengan menggunakan metode *exponentiation* dengan komputasi $P = c^{d} \pmod{n}$. Perancangan aplikasinya dapat dilihat pada *flowchart* berikut :



Gambar 7 *Flowchart* Proses Dekripsi

F. Rancangan Interface Pembangkit Kunci

Perancangan antarmuka ini adalah *form* utama pada saat aplikasi pertama sekali dijalankan. Rancangan antarmuka *form* ini dapat dilihat pada gambar berikut:



Gambar 8 Rancangan *Interface* Pembangkit Kunci

Keterangan :

1. Pembangkit Kunci : jika pembangkit kunci di klik maka aplikasi akan menampilkan halaman pembangkit kunci
2. Enkripsi : jika ini di klik maka aplikasi akan menampilkan halaman enkripsi
3. *Header* : animasi *header* untuk aplikasi
4. Dekripsi : jika ini di klik maka aplikasi akan menampilkan halaman dekripsi
5. Kata : teks kata akan digunakan untuk membangkitkan kunci *public* dan kunci *private*
6. *Submit* : jika tombol *Submit* di klik akan memproses pembangkitan kunci *public* dan kunci *private*
7. p : menampilkan nilai p untuk memproses *plainteks* ke *chiperteks*
8. q : menampilkan nilai q untuk memproses *plainteks* ke *chiperteks*

9. n : menampilkan nilai q untuk memproses *plainteks* ke *chiperteks*
10. *Totient* : menampilkan nilai totient untuk memproses *plainteks* ke *chiperteks*
11. e : menampilkan nilai e untuk memproses *plainteks* ke *chiperteks*
12. Kunci *Public* : menampilkan kunci *public* untuk memproses *plainteks* ke *chiperteks*
13. Kunci *Private* : menampilkan kunci *private* utuk memproses *chiperteks* ke *plainteks*

G. Rancangan Interface Enkripsi

Perancangan antar muka ini adalah *form* yang digunakan untuk melakukan enkripsi *plainteks* menjadi *chiperteks*, Rancangan antarmuka *form* ini dapat dilihat pada gambar berikut:

Gambar 9 Rancangan *Interface* Enkripsi

Keterangan :

1. Pembangkit Kunci : jika pembangkit kunci di klik maka aplikasi akan menampilkan halaman pembangkit kunci
2. Enkripsi : jika ini di klik maka aplikasi akan menampilkan halaman enkripsi
3. *Header* : animasi *header* untuk aplikasi
4. Dekripsi : jika ini di klik maka aplikasi akan menampilkan halaman dekripsi
5. Kunci Public : teks kunci publik yang akan digunakan untuk proses enkripsi.
6. *Browse* : jika tombol *browse* di klik akan menampilkan dialog untuk memilih berkas pesan yang akan dienkripsi
7. *Plainteks* : teks yang menampilkan pesan *plainteks* yang akan dienkripsi
8. *Chiperteks* : teks yang menampilkan pesan *chiperteks* hasil proses enkripsi
9. *Download* : jika tombol *download* di klik akan mengunduh berkas berformat Txt yang berisikan pesan *chiperteks*.

H. Rancangan Interface Dekripsi

Perancangan antar muka ini adalah *form* yang digunakan untuk melakukan dekripsi *chiperteks* menjadi *plainteks*, Rancangan antarmuka *form* ini dapat dilihat pada gambar berikut:

Gambar 10 Rancangan *Interface* Dekripsi

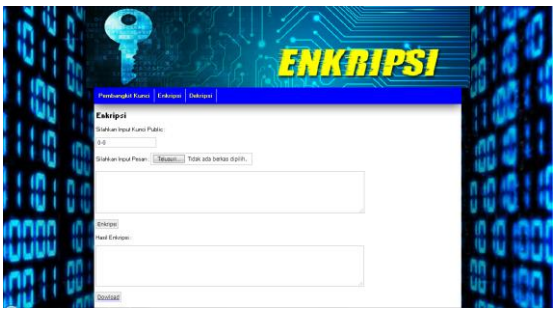
Keterangan :

1. Pembangkit Kunci : jika pembangkit kunci di klik maka aplikasi akan menampilkan halaman pembangkit kunci
2. Enkripsi : jika ini di klik maka aplikasi akan menampilkan halaman enkripsi
3. *Header* : animasi *header* untuk aplikasi
4. Dekripsi : jika ini di klik maka aplikasi akan menampilkan halaman dekripsi
5. Kunci Private : teks kunci privat yang akan digunakan untuk proses dekripsi.
6. *Browse* : jika tombol *browse* di klik akan menampilkan dialog untuk memilih berkas pesan yang akan di dekripsi
7. *Chiperteks* : teks yang menampilkan pesan *chiperteks* yang akan di dekripsi
8. *Plainteks* : teks yang menampilkan pesan *plainteks* hasil proses dekripsi
9. *Download* : jika tombol *download* di klik akan mengunduh berkas berformat Txt yang berisikan pesan plainteks .

I. Tampilan Interface Pembangkit Kunci

Gambar 11 *Interface* Pembangkit Kunci

J. Tampilan Interface Enkripsi



Gambar 12 *Interface* Enkripsi

K. Tampilan Interface Dekripsi

Gambar 13 Interface Dekripsi

V. Kesimpulan dan Saran

A. Kesimpulan

Dari pengujian bab-bab sebelumnya maka peneliti dapat memberikan kesimpulan bahwa:

1. Kombinasi Algoritma LCG (*Linear Congruential Code for Information)* sebagai pembangkit bilangan acak dan algoritma *The Sieve of Eratosthenes* sebagai pembangkit bilangan prima dapat digunakan sebagai pembangkit *public* dan *private* pada perancangan kriptografi RSA.
2. Kunci *public* dan *private* hanya dapat dibangkitkan apabila bilangan acak yang dibangkitkan dengan LCG (*Linear Congruential Code for Information)* menghasilkan lebih dari satu bilangan prima.
3. Dikarenakan *Chiperteks* merupakan hasil dari kombinasi *plainteks* dan kunci maka *file Chiperteks* yang merupakan hasil enkripsi memiliki kapasitas lebih besar dibandingkan *file plainteks.*

B. Saran

1. Dalam penelitian lebih lanjut, dapat memunculkan bilangan acak yang bersifat prima yang di gunakan untuk kunci *public* dan *private* pada aplikasi kriptografi RSA.
2. Dalam perancangan aplikasi pengamanan data, khususnya kriptografi RSA dapat merancang aplikasi yang dapat melakukan pengamanan data tanpa merubah kapasitas *file*.
3. Untuk pengembangan lebih lanjut, Aplikasi ini dapat dikembangkan untuk mengamankan *file* berformat Doc, Xls, *database* ataupun dalam bentuk format lainnya.
4. Aplikasi kriptografi *RSA* yang telah dibuat oleh peneliti diharapkan digunakan dan dimanfaatkan sebagai keamanan data

DAFTAR PUSTAKA

[1] Haro, Gok Asido. 2011. *Algoritma Pencarian Bilangan Prima*. Bandung.

[2] Madcoms, 2016. *Pemrograman PHP dan MySQL Untuk Pemula*. Yogyakarta: Andi.

[3] Munir, Rinaldi. 2014. *Sistem Kriptografi Kunci-Publik*.Yogyakarta.

[4] Sadikin, Rifki. 2012. *Kriptografi untuk Keamanan Jaringan.* Yogyakarta:Andi

[5] William 2011. *Aplikasi Enkripsi Pesan Pada iOS Menggunakan Algoritma Kriptografi Klasik Yang Diperbaharui*. Bandung.