IMPLEMENTASI KRIPTOGRAFI DENGAN ALGORITMA BLOWFISH DAN RIVERST SHAMIR ADLEMAN (RSA) UNTUK PROTEKSI FILE

Faizal Zuli[1], Ari Irawan[2]

[1]Universitas Satya Negara Indonesia, [2]UIN Syarif Hidayatullah
e-mail : [1]faizal.zuli@yahoo.com, [2]ari.irawan@uinjkt.ac.id

ABSTRAK

Blowfish merupakan salah satu algoritma yang tidak dipatenkan dan cukup kuat karena memiliki ruang kunci yang besar dan panjangnya bisa beragam sehingga tidak mudah diserang pada bagian kuncinya. Suatu sistem kriptografi yang baik terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan. RSA (RivestShamirAdleman) adalah singkatan dari para pembuatnya yaitu RonRivest, Adi Shamir, dan Leonard Adleman yang dibuat pada tahun1977. RSA merupakan sistem kriptografi asimetrik.

Kata kunci: Blowfish, Kriptografi, RSA

1. PENDAHULUAN

Untuk melindungi file yang terdapat pada komputer atau tempat penyimpanan data dari akses illegal salah satu caranya adalah dengan menggunakan kriptografi.

Kriptografi juga diperlukan dalam melindungi dokumen dari orang yang tidak berhak untuk merubah isi dokumen, merubah password atau pemanfaatan dokumen tersebut untuk keuntungan pribadi.

Kriptografi akan merahasiakan informasi dengan menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Saat ini banyak bermunculan algoritma kriptografi yang terus dianalisis, dicoba dan disempurnakan untuk mencari algoritma yang dianggap memenuhi standar keamanan. Beberapa algoritma kriptografi yang dikenal antara lain DES, Rijndael, Blowfish, RC4, Vigenere Cipher, Enigma, IDEA dan lainnya.

2. LANDASAN TEORI

2.1 Aplikasi

Aplikasi atau juga disebut program aplikasi adalah program yang dibuat oleh pemakai yang ditujukan hanyauntuk melakukan suatu tugas khusus (Kadir, 2012).

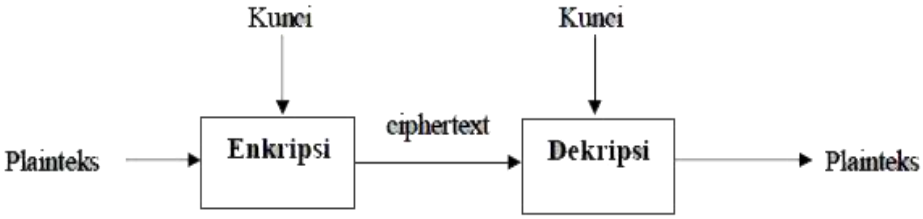
2.2 Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu cryptós yang artinya “secret” (yang tersembunyi) dan gráphein yang artinya “writting” (tulisan). Jadi, kriptografi berarti ”secret writting” (tulisan rahasia). Definisi yang dikemukakan oleh Bruce Schneier (1996), kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (Cryptography is the art and science of keeping messages secure).

Kriptografi mempunyai beberapa tujuan. Munir (2006:9) menyampaikan tujuan kriptografi bahwa untuk memberi layanan keamanan (yang juga dinamakan sebagai aspek-aspek keamanan) sebagai berikut:

a) **Kerahasiaan (*confidentiality*)**, adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapt dibaca oleh pihak-pihak yang tidak berhak.Di dalam kriptografi, layanan ini direalisasikan dengan menyandikan pesan menjadi chiperteks.
b) **Integritas data (*data integrity*)**, adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.Dengan kata lain, aspek keamanan ini dapat diungkapkan sebagai pernyataan: “Apakah pesan yang diterima masih asli atau tidak mengalami perubahan (modifikasi)?”. Untuk menjaga integritas data, system harus memiliki kemampuan untuk mendeteksi manipulasi pesan oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubstitusian data lain ke dalam pesan yang sebenarnya.

Kriptografi mempunyai dua komponen utama yaitu enkripsi dan dekripsi. Selainitu dibutuhkan kunci untuk mengubah plainteks menjadi chiperteks dan juga sebaliknya.Tanpa kunci plainteks tidak bisa mengenkrip masukan menjadi chiperteks, demikianjuga sebaliknya. Kerahasiaan kunci sangatlah penting, apabila kerahasiaannya terbongkar maka isi pesan akan terbongkar. Berikut adalah skema yang mengilustrasikan enkripsi dan dekripsi:



Gambar 1 Skema Enkripsi dan Dekripsi

2.3 Algoritma *Blowfish*

Blowfish diciptakan oleh seorang Cryptanalyst bernama Bruce Schneier, Presiden perusahaan Counterpane Internet Security, Inc (Perusahaan konsultan tentang kriptografi dan keamanan komputer) dan dipublikasikantahun 1994. Dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengancache data yang besar). Blowfish merupakan algoritma yang tidak dipatenkan dan licensefree, dan tersediasecara gratis untuk berbagai macam kegunaan (Syafari, 2007). Pada saat blowfish dirancang, diharapkan mempunyai kriteria perancangan sebagai berikut (Schneier, 1996):

a) Cepat, Blowfish melakukan enkripsi data pada microprocessors 32-bit dengan rate 26 clock cycles perbyte.
b) Ringan (*compact*), Blowfish dapat dijalankan pada memori kurang dari 5K.
c) Sederhana, Blowfish hanya menggunakan operasi-operasi sederhana: penambahan, XOR, dan lookuptabel pada operan 32-bit.
d) Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh Blowfish dapatbervariasi dan bisa sampai sepanjang 448 bit.

Algoritma *Blowfish* terdiri atas dua bagian, yaitu ekspansi kunci dan enkripsi data (Schneier, 1996).

- **Ekspansi Kunci (*Key-expansion*)**
  Berfungsi merubah kunci (minimum 32-bit, maksimum 448-bit)menjadi beberapa array subkunci (subkey)dengan total 4168 byte (18x32-bit untuk P-array dan 4x256x32-bit untuk S-box sehingga totalnya 33344 bit atau4168 byte). Kunci disimpan dalam K-array:

  $$K1, K2, \ldots Kj \; 1 \leq j \leq 14$$

  Kunci-kunci ini yang dibangkitkan (*generate*) dengan menggunakan subkunci yang harus dihitung terlebih dahulu sebelum enkripsi atau dekripsi data. Sub-sub kunci yang digunakan terdiri dari:P-array yang terdiri dari 18 buah 32-bit subkunci.

- **Enkripsi Data**
  Terdiri dari iterasi fungsi sederhana (*Feistel Network*) sebanyak 16 kali putaran (iterasi), masukannya adalah 64-bit elemen data X. Setiap putaran terdiri dari permutasi kunci-dependent dan substitusi kunci dan data-*dependent*. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32-bit. Operasi tambahan lainnya.

2.4 Algoritma RSA (Riverst Shamir Adleman)

Algoritma ini dikembangkan oleh Ron Rivest, Adi Shamir, dan Len Adleman pada tahun 1977. Algoritma ini sekaligus menjawab tantangan dari sebuah paper yang dibuat oleh Diffie dan Hellman tentang pendekatan baru mengenai algoritma kriptografi yang dapat memenuhi kebutuhan untuk metode kunci publik. Algoritma Rivest Shamir Adleman (RSA) ini adalah algoritma metode kunci publik yang paling banyak dipakai sampai saat ini. Cara kerja dari algoritma RSA ini adalah sebagai berikut:

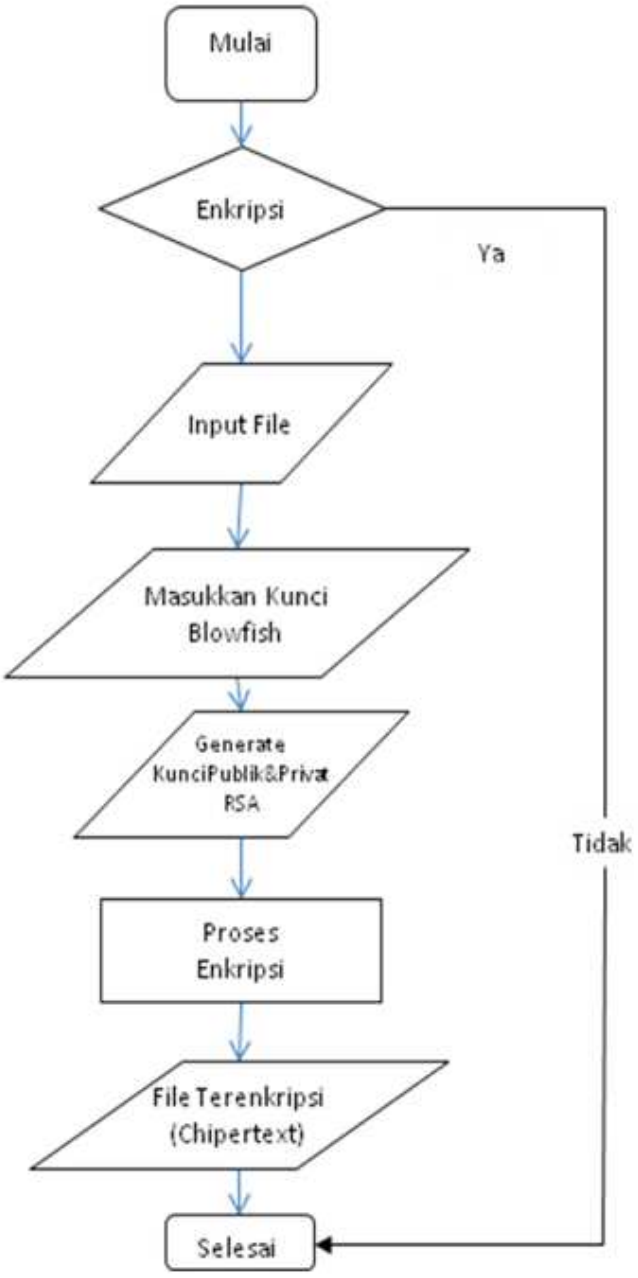
- RSA merupakan algoritma yang melibatkan ekspresi dengan fungsi eksponensial.
- Plaintext dienkripsi dalam blokblok, dimana setiap blok tersebut mempunyai nilai biner yang kurang dari angka tertentu (*n).*
- Proses enkripsi dan dekripsi untuk plaintext blok M dan ciphertext blok C dapat digambarkan sebagai berikut:
  C = Me mod *n*
  M = Cd mod *n* = (Me) d mod n = Med mod n

3. HASIL DAN PEMBAHASAN

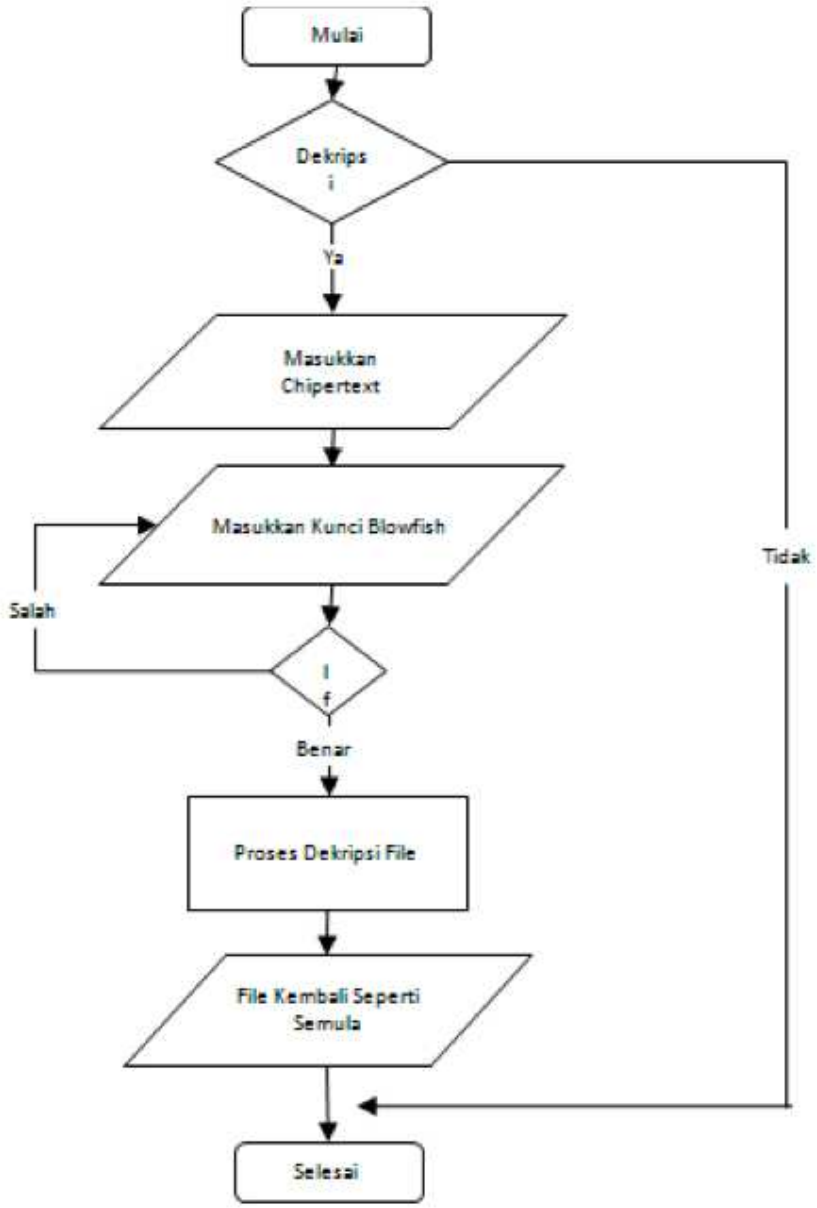
3.1 Enkripsi File

Proses enkripsi file yang dilakukan adalah sebagai berikut:

1) Pengguna memasukkan input berupa *file*. File yang akan diinputkan berupa *file* teks.
2) Masukkan kunci *blowfish* untuk mengenkripsi.
3) Lakukan enkripsi *file* yang telah diinputkan.
4) *File* yang telah terenkripsi menjadi *file* yang tidak terbaca (chipertext).
5) Chipertext dienkripsi kembali dengan algoritma RSA.
6) RSA kunci generator (kunci publik dan kunci privat).
7) Enkripsi file selesai



Gambar 2 Enkripsi File



Gambar 3 Dekripsi File

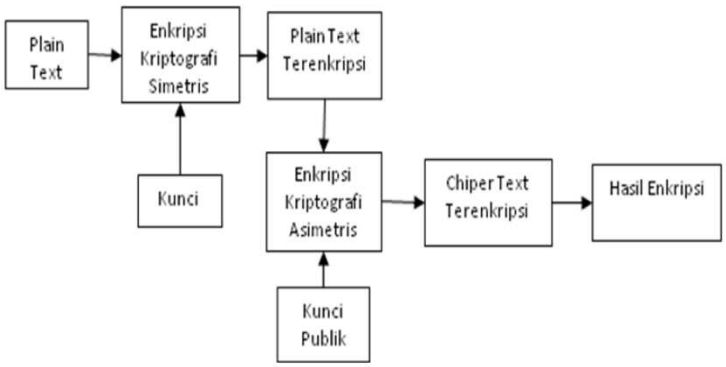
3.2 Melakukan Pembangkitan Kunci Publik dan Kunci Privat

1) Ambil dua buah bilangan prima sembarang, x dan y
2) Hitung n = xy dan m = (x-1)*(y-1).
3) Pilih bilangan integer e, 1 < e < m, yang relative prima terhadap m yaitu FPB(e,m) = 1.
4) Hitung eksponen rahasia d, 1 < d < m, sehingga ed ≡1 (mod m).
5) Kunci publik adalah (n,e) dan kunci privat adalah (n,d). Nilai x, y dan m juga harus dirahasiakan.
   - n disebut juga modulus
   - e disebut juga public exponent atau exponent enkripsi
   - d disebut juga secret exponent atau exponent dekripsi

3.3 Melakukan Enkripsi Kunci Simetris dan Plainteks Terenkripsi dengan Enkripsi Kriptografi Asimetris RSA

Adapun langkah-langkah pada proses ini adalah sebagai berikut:

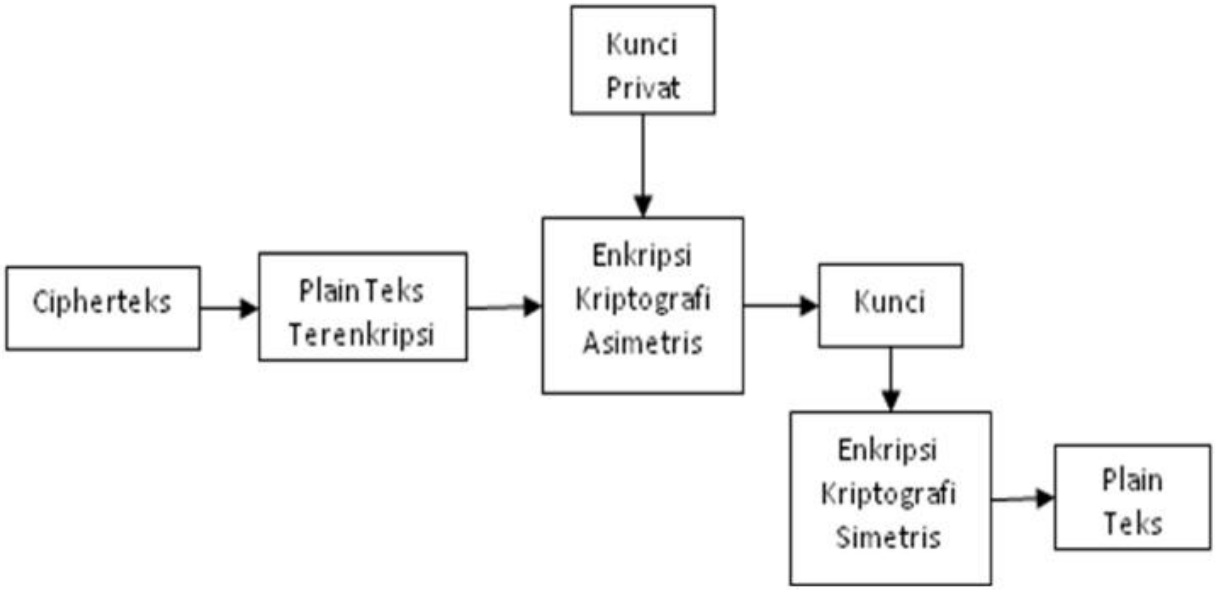
1) Ambil kunci public (n,e)
2) Nyatakan plainteks dalam integer positif m
3) Enkripsi menjadi cipher teks $c = m^e \bmod n$



Gambar 4 Enkripsi dengan Kunci Publik

3.4 Melakukan Dekripsi Kunci Simetris dan Plainteks Terenkripsi dengan Dekripsi Kriptografi Asimetris RSA

Menggunakan kunci privat (n, d) untuk dekripsi dengan rumus $m = c^d \bmod n$.

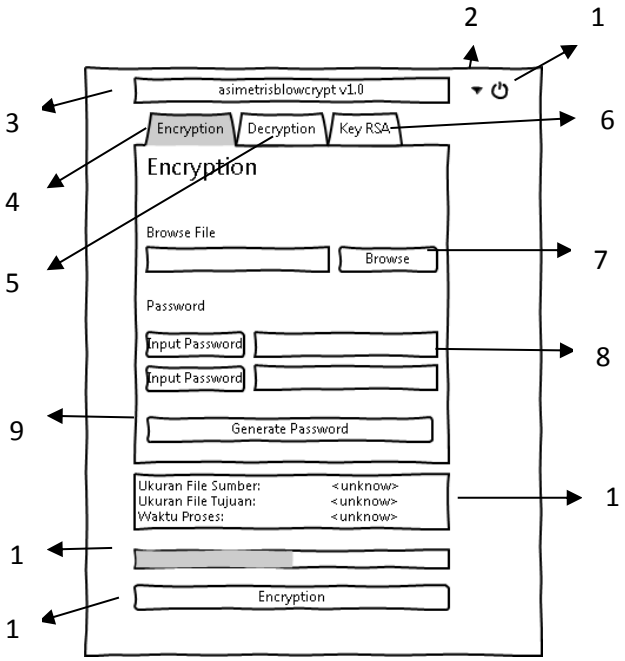


Gambar 5 Dekripsi Dengan Kunci Privat

4. PERANCANGAN TAMPILAN ANTARMUKA

Pada perancangan tampilan antarmuka aplikasi asimetrisblowcrypt v1.0, dibangun dengan java untuk menampilkan tampilan antarmuka yang nyaman dan mudah digunakan bagi pengguna.

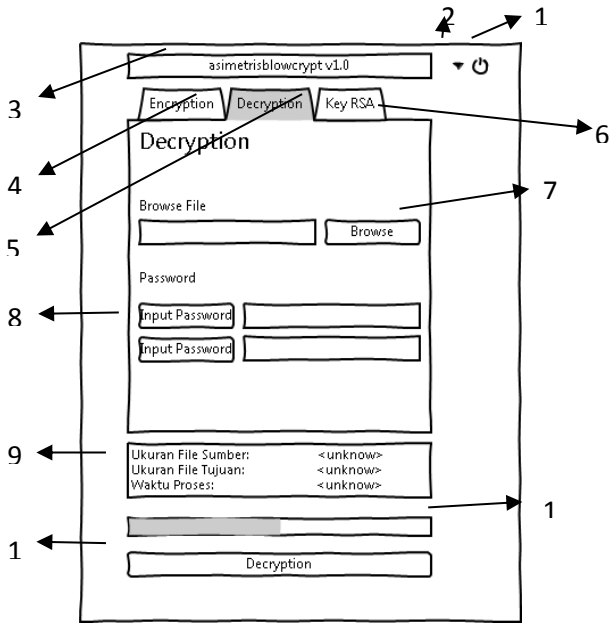
- Halaman Enkripsi



Gambar 6 Tampilan Antarmuka Halaman Enkripsi

Keterangan gambar:
1. Tombol untuk menutup aplikasi.
2. Tombol untuk *minimize* aplikasi.
3. Nama aplikasi dan versi aplikasi.
4. Tombol tab menu enkripsi.
5. Tombol tab menu dekripsi.
6. Tombol tab menu *Key* RSA
7. Tombol *browse file*.
8. Tombol *input password.*
9. Tombol *generate password*
10. Informasi ukuran *file* sumber, ukuran *file* tujuan, dan waktu proses.
11. *Progress bar*.
12. Tombol *Encryption*

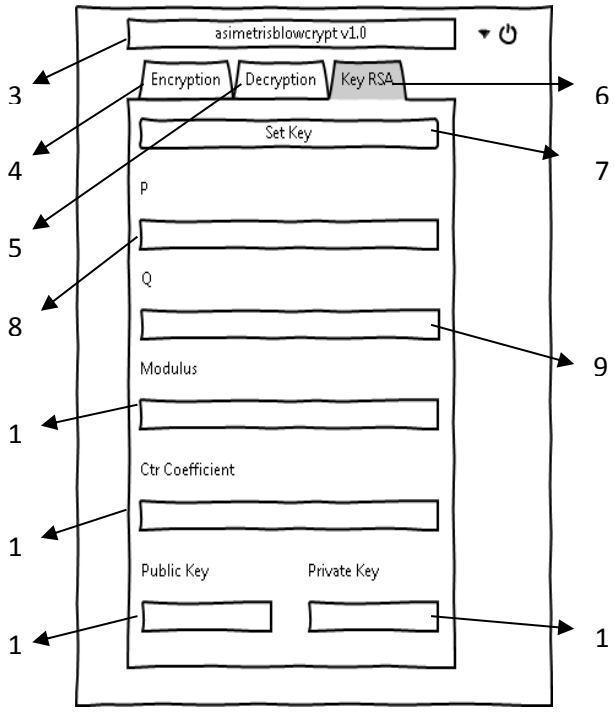
- Halaman Dekripsi



Gambar 7 Tampilan Antarmuka Halaman Dekripsi

Keterangan gambar:
1. Tombol untuk menutup aplikasi.
2. Tombol untuk *minimize* aplikasi.
3. Nama aplikasi dan versi aplikasi.
4. Tombol tab menu enkripsi.
5. Tombol tab menu dekripsi.
6. Tombol tab menu *Key* RSA
7. Tombol *browse file*.
8. Tombol *input password.*
9. Informasi ukuran *file* sumber, ukuran *file* tujuan, dan waktu proses.
10. *Progress bar*.
11. Tombol *Decryption*.

- Key RSA
  Berikut adalah rancangan halaman input key RSA:



Gambar 8 Tampilan Antarmuka Halaman Key RSA

Keterangan gambar:
1. Tombol untuk menutup aplikasi.
2. Tombol untuk *minimize* aplikasi.
3. Nama aplikasi dan versi aplikasi.
4. Tombol tab menu enkripsi.
5. Tombol tab menu dekripsi.
6. Tombol tab menu *Key* RSA.
7. Tombol *Set Key*.
8. Hasil *generate* P.
9. Hasil *generate* Q.
10. Hasil *generate modulus*.
11. Hasil *generate ctr coefficient*.
12. Hasil *generate Public Key*.
13. Hasil *generate Private Key*.

5. PERANCANGAN TAMPILAN ANTARMUKA

- Perangkat Lunak:
  - Java (SE) Run Time Environment1.8.0_25.
  - Netbeans IDE 7.4.
  - Adobe Photoshop.
  - Visual Paradigm For UML.
  - Advance Installer 12.5

- Perangkat Keras:
  - Notebook MSI FX400 dengan spesifikasi Intel® Core™ i3-350M
  - Processor (2.26 GHz, Cache 3MB)

- RAM 4 GB DDR3 SODIMM PC-8500
- VGA NVIDIA GeForce GT325M 1GB
- Kamera onboard
- Resolusi layar 1366 x 768
- Kapasitas Penyimpanan 320GB.

6. KESIMPULAN

Dari analisa algoritma dan implementasi program, maka hasil yang didapatkan dengan menggunakan algoritma Blowfish dan RSA dapat disimpulkan sebagai berikut:

1) Enkripsi dengan algoritma *blowfish* terdiri dari dua bagian, yaitu ekspansi kunci yang berfungsi merubah kunci (minimum 32-bit, maksimum 448-bit) menjadi beberapa array subkunci (subkey) dengan total 4168 *byte* (18x32-bit untuk *P-array* dan 4x256x32-bit untuk S-box sehingga totalnya 33344 bit atay 4168 *byte*) dan enkripsi data.
2) Enkripsi dengan algoritma *RSA* memiliki dua buah kunci yang berbeda, yaitu kunci publik dan kunci privat. Prinsip kerja algoritma *RSA* menggunakan operasi pemangkatan dan operasi mod (modulus) yang menghasilkan nilai yang relatif acak.
3) Ukuran *file* sebelum dan sesudah dienkripsi dengan algoritma *Blowfish* dan *RSA* terjadi perubahan karena penambahan ukuran *file* sesuai dengan algoritma yang digunakan.

DAFTAR PUSTAKA

[1] Ariyus, Dony. 2006. Kriptografi: Keamanan Data dan Komunikasi**.** Yogyakarta : Graha Ilmu
[2] Munir, Rinaldi. 2006. Kriptografi. Bandung : Informatika.
[3] Hakim S, Rachmad. 2010. Buku Pintar Windows 7. Jakarta: PT Elex Media Komputindo
[4] Pressman, Roger. 2005**.** *Software Engineering: A Practitioner's Approach,* Edisi ke 6. New York: McGraw-Hill
[5] Kadir, Abdul. 2012. Algoritma & Pemrograman Menggunakan Java. Jakarta: Andi