Implementasi Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA	**: NOVIANA ASTUTI**
NPM	**: 1514370002**
PROGRAM STUDI	**: SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

NOVIANA ASTUTI

Implementasi Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)

2019

Perancangan aplikasi keamanan dokumen ini bertujuan untuk mengamankan sebuah dokumen yang telah dibuat agar tidak diubah isinya dan keasliannya hilang. Aplikasi berbasis desktop ini dirancang dengan pemodelan UML (*Unified Modelling Language*) dan menggunakan metode algoritma kriptografi RSA (Rhivest Shamir Adleman) untuk proses enkripsi dan dekripsi. Dalam perancangan aplikasi menggunakan UML yang dibuat antara lain: *use case diagram, activity diagram, class diagram, sequence diagram, deployment diagram*, diagram objek, diagram status, dan diagram komponen. Dalam tahap *coding*, software yang digunakan adalah Visual Basic 2010. Dokumen yang telah diketik dan disimpan baik itu di komputer sendiri ataupun disimpan di *flashdisk* yang bisa jatuh dan hilang yang dapat ditemukan orang lain, dokumen menjadi tidak aman karena bisa diubah keasliannya dan disebarkan dengan data palsu atas nama orang sebelumnya. Aplikasi ini diharapkan mampu meminimalisir kemungkinan buruk yang terjadi.

Kata Kunci : *UML, kriptografi, RSA, enkripsi, dekripsi*

DAFTAR ISI

Halaman

BAB III

Metode Penelitian

3.1 Tahapan Penelitian

Pada bagian ini dijelaskan tahapan atau cara - cara memperoleh data - data yang digunakan untuk kebutuhan penelitian. Untuk memudahkan peneliti dan pembaca memahami penelitian, maka akan lebih baik dibuat tahapan dalam bentuk *flow chart* (Lampiran 10).

3.2 Metode Pengumpulan Data

Dalam pengumpulan data, peneliti menempuh langkah-langkah melalui penelitian kepustakaan (*library research*) yaitu suatu penelitian kepustakaan murni. Metode riset ini dipakai untuk mengkaji sumber-sumber tertulis.

3.3 Analisa Sistem Yang Sedang Berjalan

Sub bab ini berisikan tentang analisa sistem yang akan dibangun. Sub bab ini membahas teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian - bagian komponen dengan tujuan mempelajari seberapa baik bagian - bagian komponen tersebut bekerja dan berinteraksi.

Bedasarkan analisa yang dilakukan oleh penulis terhadap sistem yang sedang berjalan. Sistem hanya melakukan penyimpanan *file* dalam bentuk dokumen hal ini sangat rentan dengan terjadinya pencurian data yang dapat dilakukan oleh pihak-pihak yang tidak bertanggung jawab. *File* - *file* tersebut

dapat dengan mudah di curi karena tidak ada sistem yang dapat mengamankan data - data nilai tersebut.

1. Analisa Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses - proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga berisi in*form*asi-in*form*asi apa saja yang harus ada dan dihasilkan oleh sistem. Berikut kebutuhan fungsional yang terdapat pada sistem yang dibangun:

a. Mengimplementasikan penggunaan *Visual Basic.Net 2010* dalam membuat aplikasi sistem keamanan data pada file Microsoft office dengan menggunakan algoritma RSA.
b. Aplikasi harus dapat melakukan enkripsi terhadap sebuah *file *.xls* dan **.doc*.
c. Aplikasi harus dapat melakukan dekripsi terhadap *file* yang sudah dienkripsi, tanpa merusak *file*.

2. Analisa Kebutuhan NonFungsional

Kebutuhan ini adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh sistem. Berikut adalah kebutuhan nonfungsional yang dimiliki sistem:

1. Operasional
   a. Dapat digunakan pada sistem operasi *Microsoft Windows XP/Vista/7* secara *stand alone*.
   b. Aplikasi dibangun dengan menggunakan komponen *IDE Visual Studio 2010*

c. Spesifikasi komputer standard *Processor Pentium IV* 2,6 GHz, Memori 512 MB, Kartu Grafik 128 MB

2. Kinerja

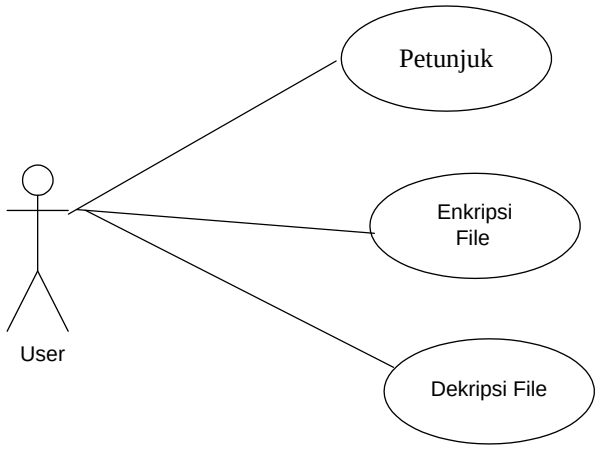
Waktu yang diperlukan dalam mengeksekusi aplikasi sistem keamanan data file Microsoft office menggunakan algoritma RSA yang dibangun cukup ringan, sehingga eksekusi tampilannya cukup cepat.

3.4 Rancangan Penelitian

Sub bab ini berisikan tentang rancangan sistem yang akan dibangun, dalam hal ini perancangan terhadap sistem.

1. Diagram *Use Case*

Berikut ini merupakan diagram *use case* dari "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.1

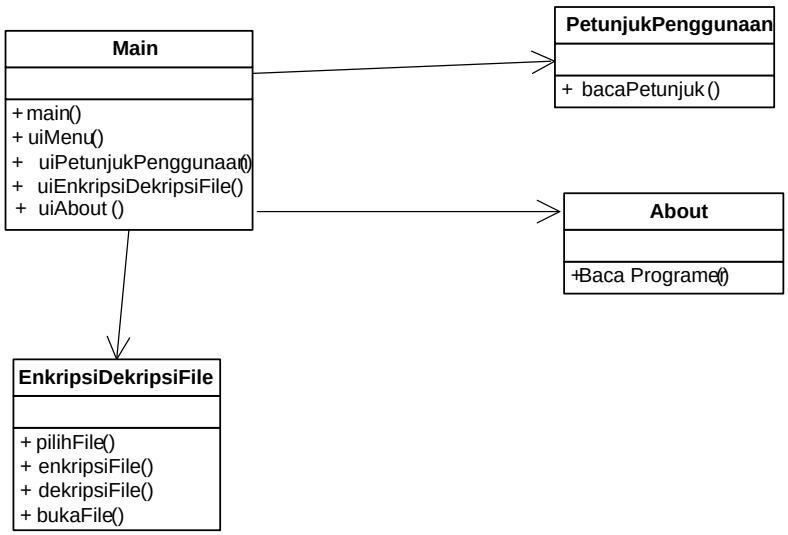


Gambar 3.1. Diagram *Use Case* Aplikasi

Pada diagram *use case* di atas, aktor yang didefenisikan pada aplikasi hanya satu, yaitu *user*. *User* adalah orang yang menjalankan aplikasi. Ketika aplikasi dijalankan, aplikasi akan menampilkan halaman dan mengeksekusi perintah sesuai dengan *event* yang diberikan *user* pada *interface* aplikasi.

2. Diagram *Class*

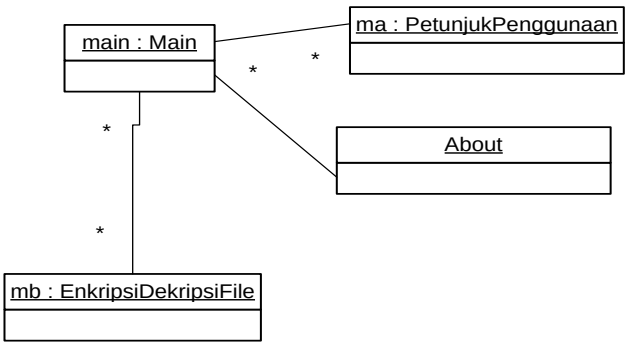
Berikut ini merupakan diagram kelas dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.2



Gambar 3.2. Diagram *Class* Aplikasi

3. Diagram Objek

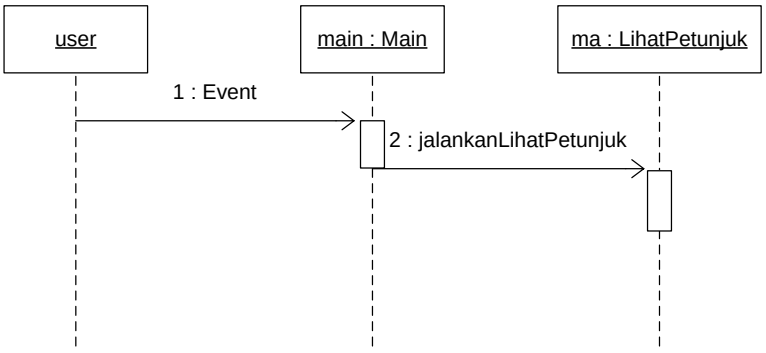
Berikut ini merupakan diagram objek dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.3



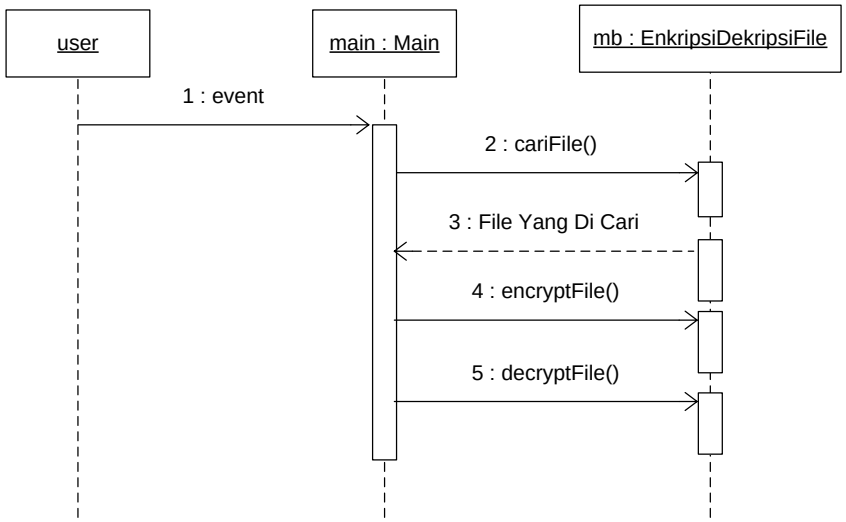
Gambar 3.3 Diagram Objek Aplikasi

4. Diagram *Sequence*

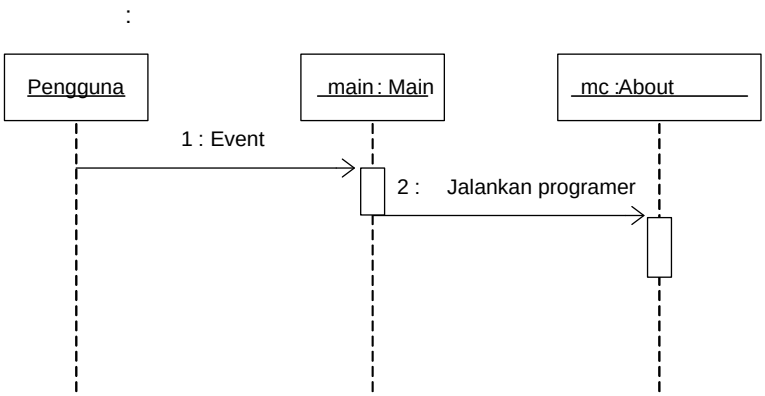
Berikut ini merupakan diagram *sequence* dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.4, dan gambar 3.5



Gambar 3.4 Diagram *Sequence* Petunjuk Penggunaan



Gambar 3.5 Diagram *Sequence* Enkripsi Dekripsi *File*

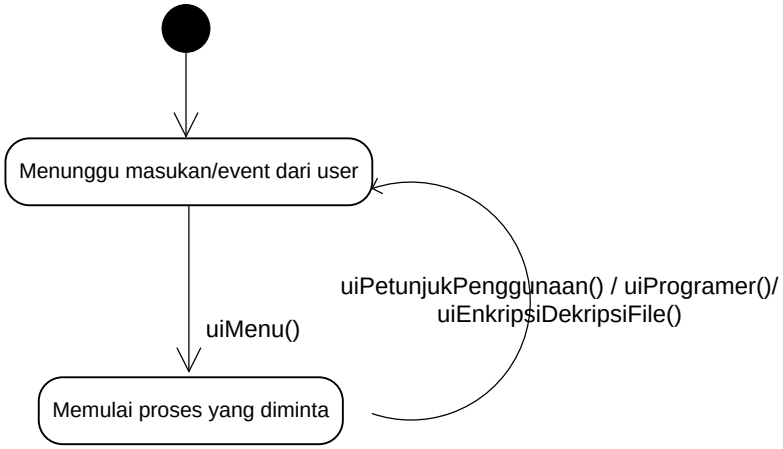


Gambar 3.6 Diagram *Sequence About*

5. Diagram Status

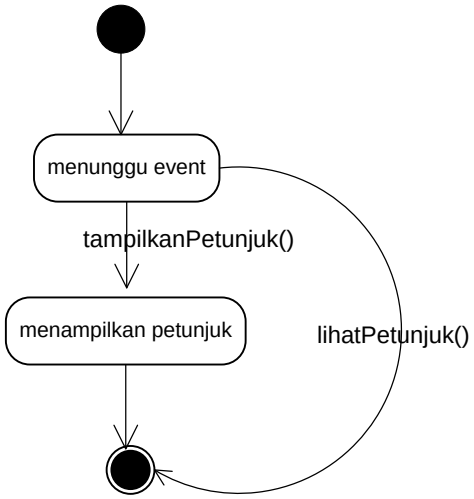
Berikut ini merupakan diagram status dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)".

Objek : main dari kelas Main



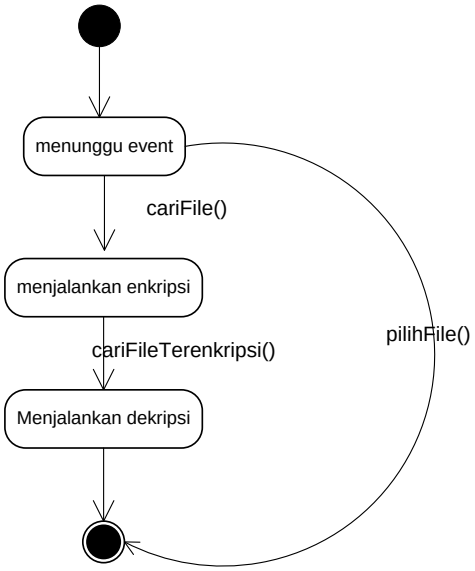
Gambar 3.7 Diagram Status Objek : main dari kelas Main

Objek : ma dari kelas PetunjukPenggunaan



Gambar 3.8 Diagram Status Objek : ma dari kelas PetunjukPenggunaan

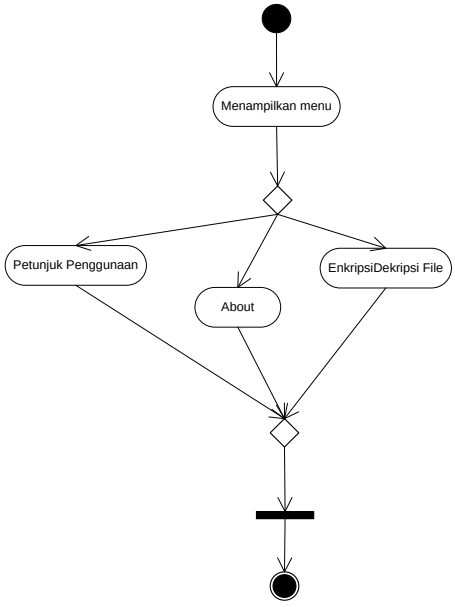
Objek : mb dari kelas EnkripsiDekripsi*File*



Gambar 3.9 Diagram Status Objek : mb dari kelas EnkripsiDekripsi *File*

6. Diagram Aktivitas

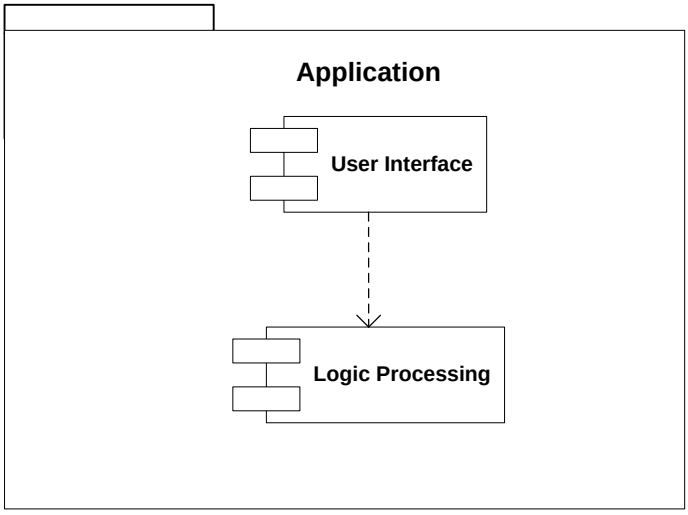
Berikut ini merupakan diagram aktivitas dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.9



Gambar 3.10 Diagram Aktivitas

7. Diagram Komponen

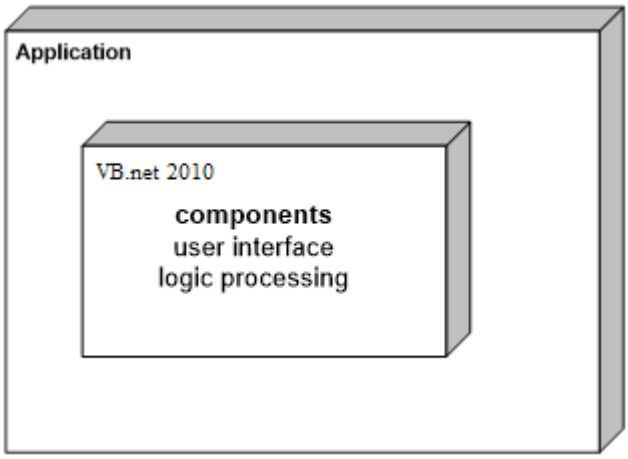
Berikut ini merupakan diagram komponen dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.11



Gambar 3.11 Diagram Komponen

8. Diagram *Deployment*

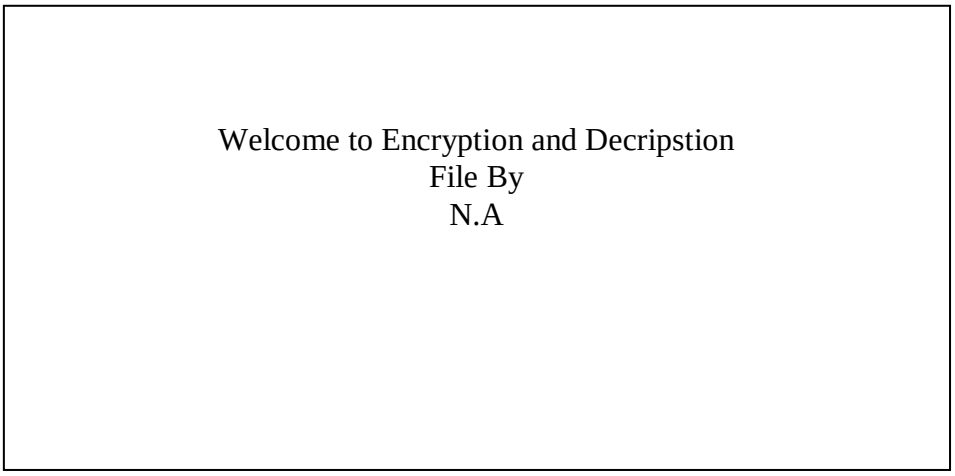
Berikut ini merupakan diagram *deployment* dari aplikasi "Rancang Bangun Sistem Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)". Terlihat pada gambar 3.12



Gambar 3.12 Diagram *Deployment*

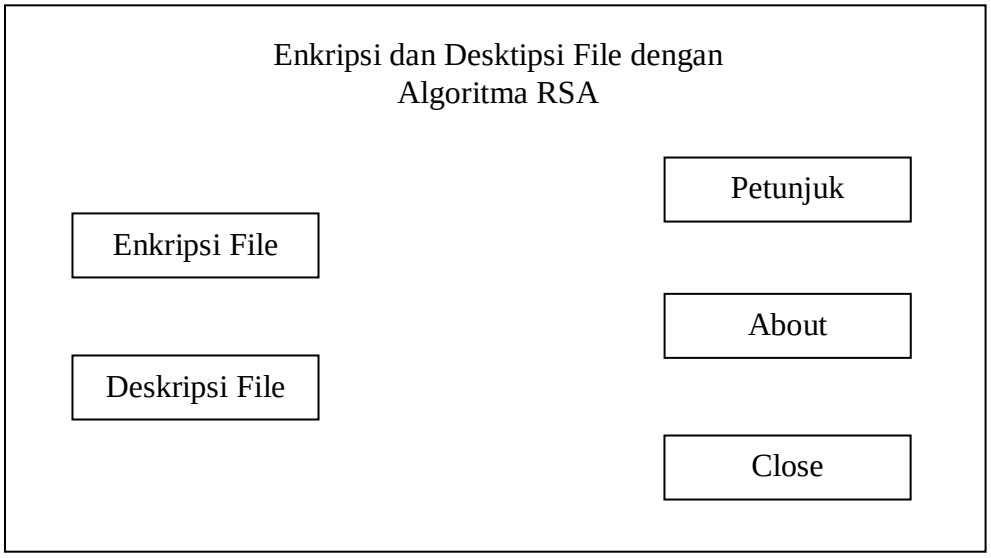
9. Perancangan Tampilan

a. Rancangan Tampilan Awal



Gambar 3.13 Rancangan *Form* Tampilan Awal

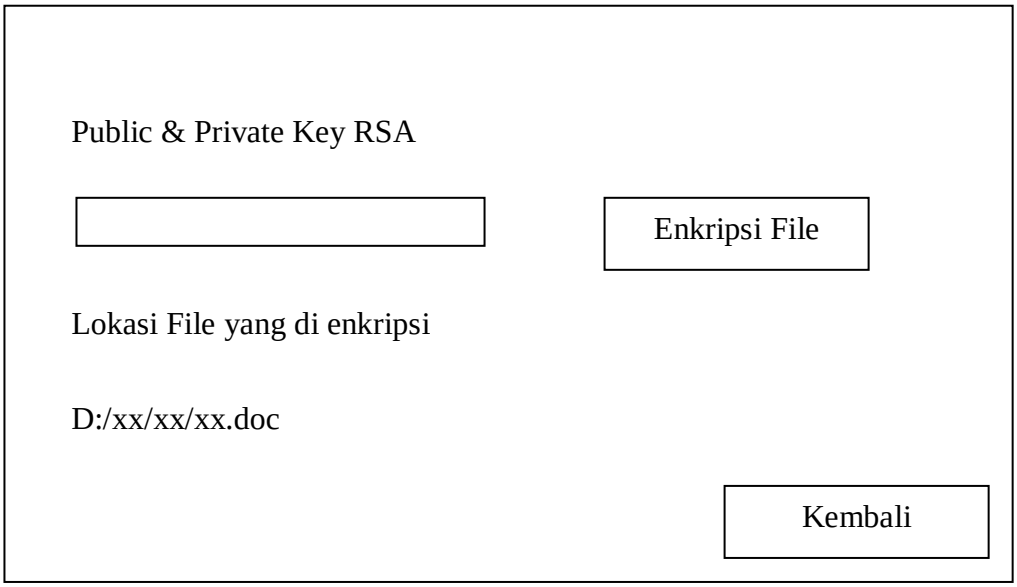
Form di atas merupakan *form* awal yang akan ditampilkan pada saat aplikasi dijalankan. Pada saat *form* awal ini tampil, teks akan berjalan dari bawah ke atas.



Gambar 3.14 Rancangan *Form* Tampilan *Menu* Pilihan

Form di atas merupakan *form* kedua yang akan ditampilkan pada saat *form* awal aplikasi dijalankan. Pada saat *form* kedua ini tampil, terdapat lima buah pilihan yang dihadapkan kepada user yakni "Petunjuk Penggunaan" untuk menuju *form* Petunjuk Penggunaan Aplikasi, "Enkripsi *File*" untuk melakukan enkripsi *file*, "Dekripsi *File*" untuk melakukan dekripsi *file*, "About" untuk menuju *form* informasi pembuat aplikasi, "Close" untuk keluar dari aplikasi.

c. Rancangan Tampilan *Form* Enkripsi *File*

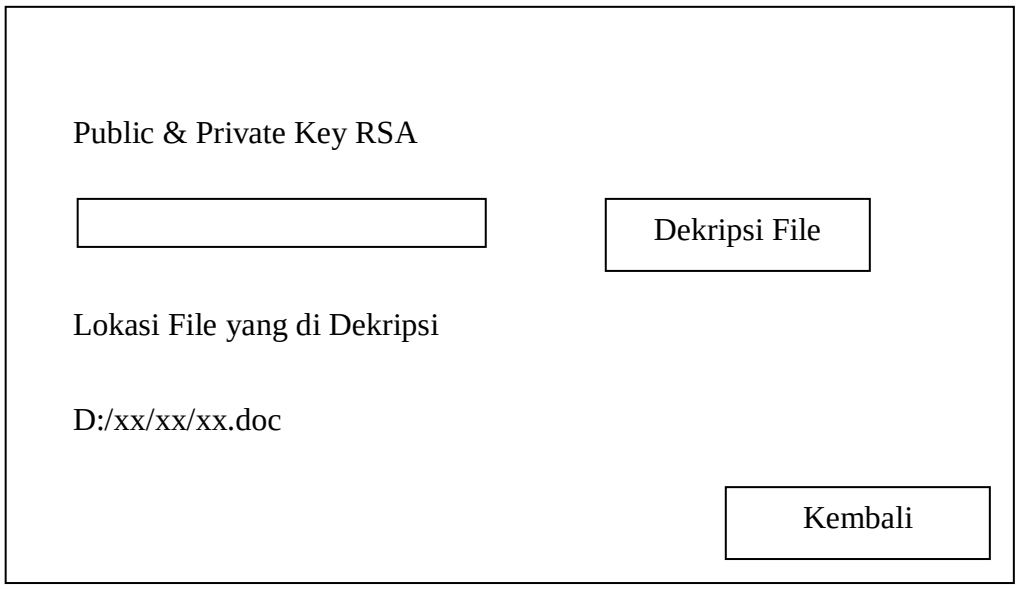


Gambar 3.15 Rancangan *Form* Enkripsi *File*

Rancangan *form* di atas merupakan *form* utama aplikasi yang berfungsi sebagai *form* untuk melakukan enkripsi *file*. Pada saat pengguna ingin melakukan enkripsi terhadap suatu *file*, pengguna harus membuat kunci pada *public* dan *private* key, kemudian pilih *file* yang akan dienkripsi dengan menekan tombol enkripsi *file*, baru kemudian dilakukan enkripsi, lihat hasil *file* yang sudah dienkripsi. Demikian juga pada saat mendekripsi

file, pengguna harus mencari *file* yang sudah terenkripsi terlebih dahulu dengan menekan tombol dekripsi *file*, baru kemudian dilakukan dekripsi, lihat hasil untuk membuka *file* yang sudah di-dekripsi.

d. Rancangan Tampilan *Form* Dekripsi *File*



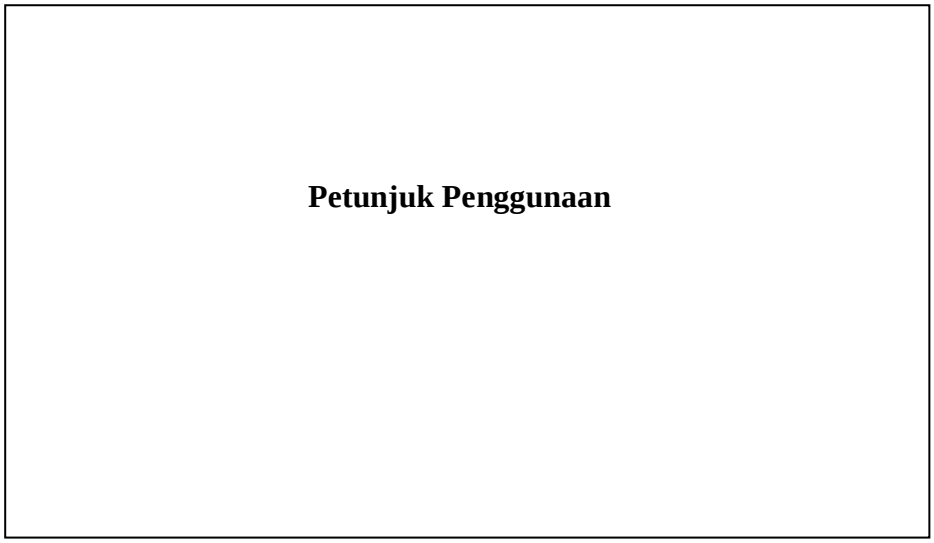
Gambar 3.16 Rancangan *Form* Dekripsi *File*

Rancangan *form* di atas merupakan *form* utama aplikasi yang berfungsi sebagai *form* untuk melakukan dekripsi *file*. Pada saat pengguna ingin melakukan dekripsi terhadap suatu *file*, pengguna harus membuat kunci pada *public* dan *private* key, pengguna harus mencari *file* yang sudah terenkripsi terlebih dahulu dengan menekan tombol dekripsi *file*, baru kemudian dilakukan dekripsi, lihat hasil untuk membuka *file* yang sudah didekripsi.

e. Rancangan Tampilan *Form* Petunjuk

Form petunjuk merupakan *form* yang berisi petunjuk penggunaan aplikasi sehingga bila ada pengguna baru akan membuka aplikasi

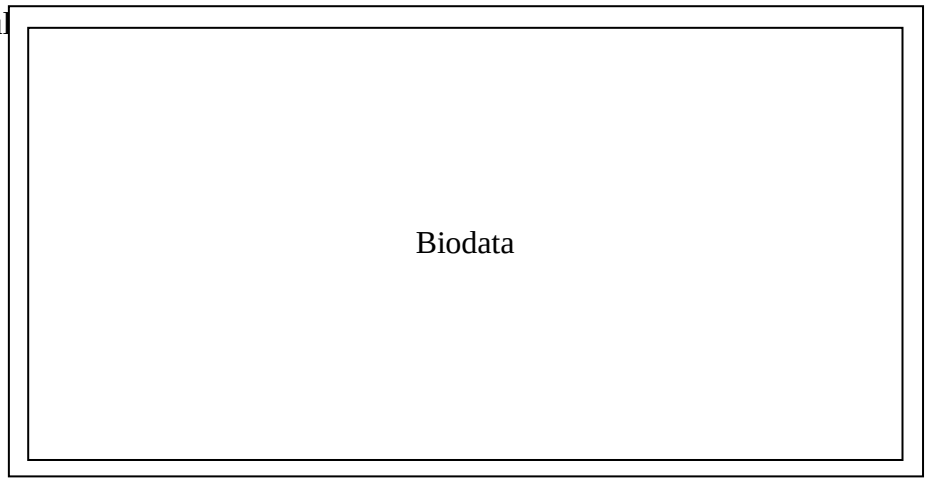
pengguna tersebut tidak perlu bertanya kepada pengguna sebelumnya. Terlihat pada gambar 3.17



Gambar 3.17 Rancangan *Form Menu* Petunjuk

f. Rancangan Tampilan *Form About*

Form about merupakan *form* yang berisi tentang biodata penul



Gambar 3.18. Perancangan *Form About*

3.4 Algoritma

1. Algoritma Pembuatan *Private key* dan *Public key*

Adapun langkah-langkah proses pembuatan *private key* dan *public key* dalam algoritma RSA adalah sebagai berikut :

a. Cari 2 Bilangan Prima secara acak.

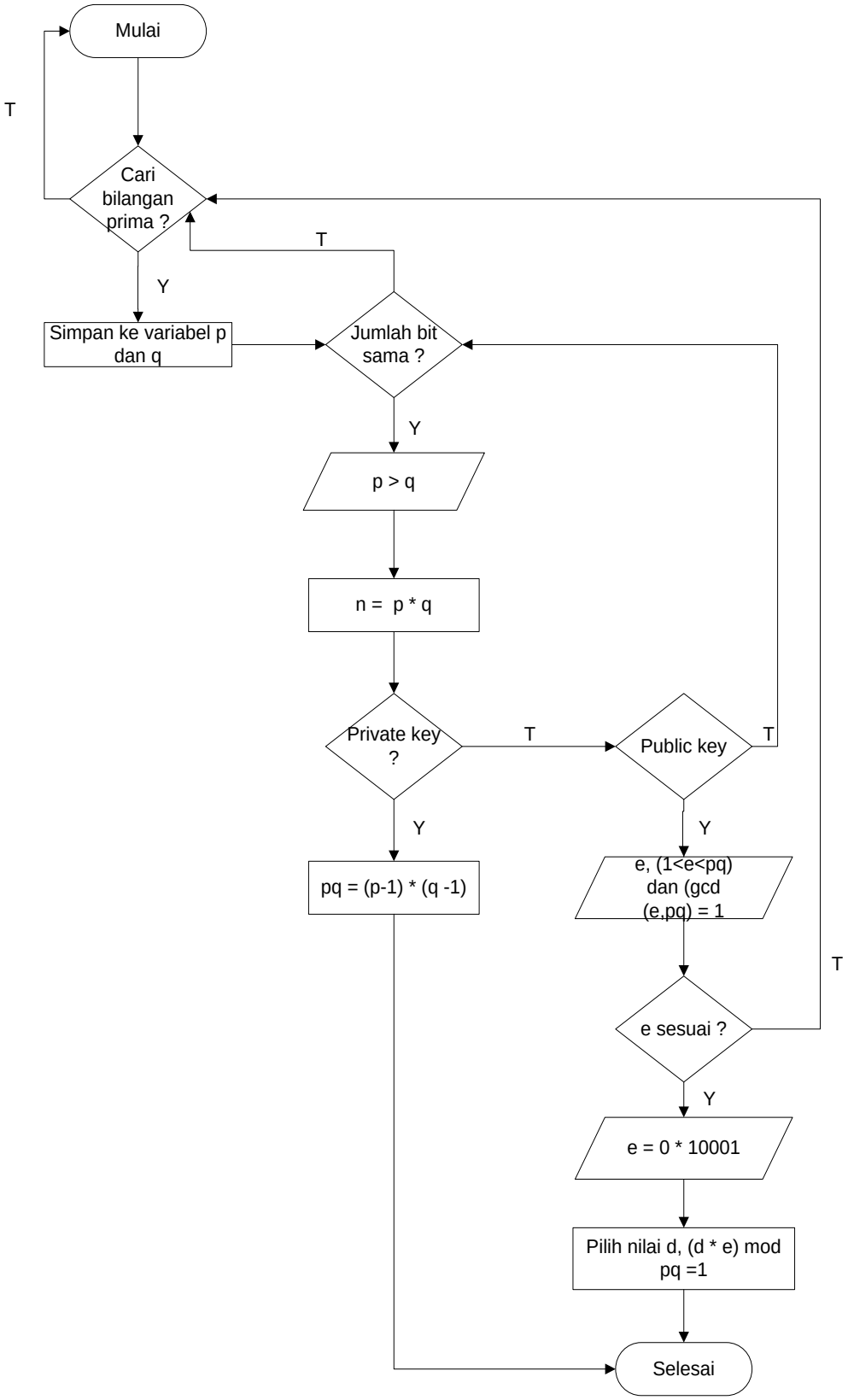
b. Simpan dalam variabel p dan q, jumlah bit untuk bilangan ini sama. Nilai p harus lebih besar dari q dan direkomendasikan minimal untuk menggunakan bilangan di atas 128bit/2 = 64bit bila akan membuat kunci dengan *bit-length* sebesar *128bit* (min 64bit hex = 0x8000000000000000; min 64bit desimal = 9223372036854775808).

c. Hitung n = p*q; Dimana nilai n ini akan digunakan untuk modulus pada *private* dan *public* key.

d. Hitung pq = (p-1) * (q-1); Untuk digunakan sebagai pencarian nilai *private* key.

e. Pilih nilai e untuk *public* key dengan syarat (1< e < pq) dan (gcd(e,pq)=1); Nilai e ini biasanya merupakan nilai yang relatif kecil, yang paling sering digunakan adalah 0x1001 = 65537. Bila kriteria e tidak cocok dengan syarat di atas, maka harus dicari nilai e lain yang sesuai, atau bila e sudah ditentukan dengan 0x10001, maka yang harus dicari kembali adalah nilai p, q, n dan pq seperti pada tahap awal.

f. Pilih nilai d, dengan syarat nilai d memenuhi : (d*e) mod pq = 1

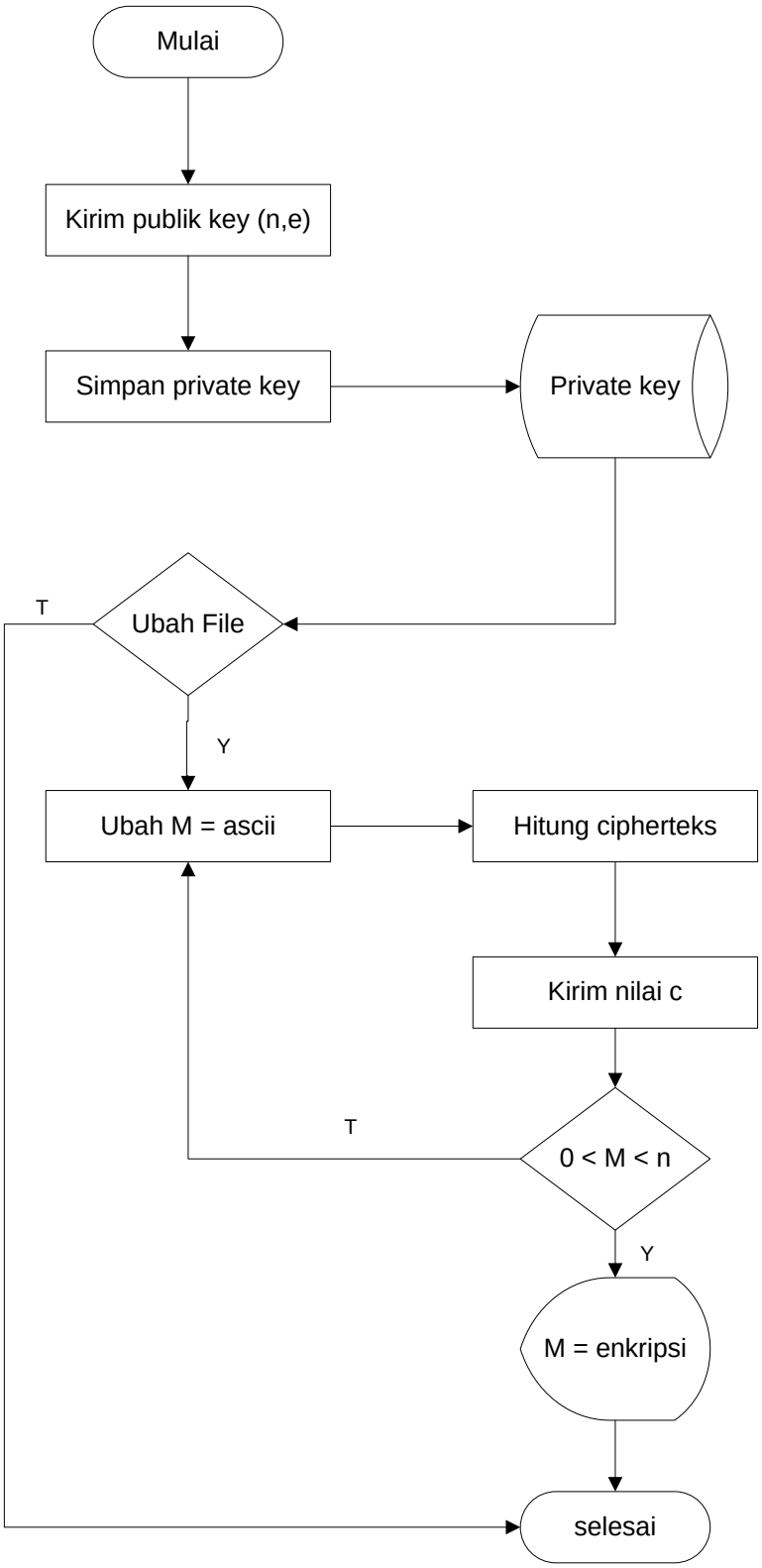


Gambar 3.19. *Flowchar*t Pembuatan *Private key* dan *Public key*

2. Algoritma Proses Enkripsi

Adapun langkah-langkah proses enkripsi dalam algoritma RSA adalah sebagai berikut :

a. Si A mengirimkan *public key* (n,e) nya untuk si B, dan menyimpan secara rahasia *private key*-nya.
b. Si B ingin mengubah *file* "M" pada si A.
c. Si B kemudian merubah “M” menjadi kode *ascii* (berupa *integer*).
d. Menghitung *ciphertext* "c" (nilai yang telah terenkripsi) dengan menggunakan *public key* yang dikirimkan oleh si A kepadanya.
e. Kemudian B mengirimkan nilai “c” kepada A untuk di-*decrypt* dengan menggunakan *private-key* miliknya.
f. Nilai M harus lebih besar dari 0, dan harus lebih kecil dari nilai n (dari *public key*).



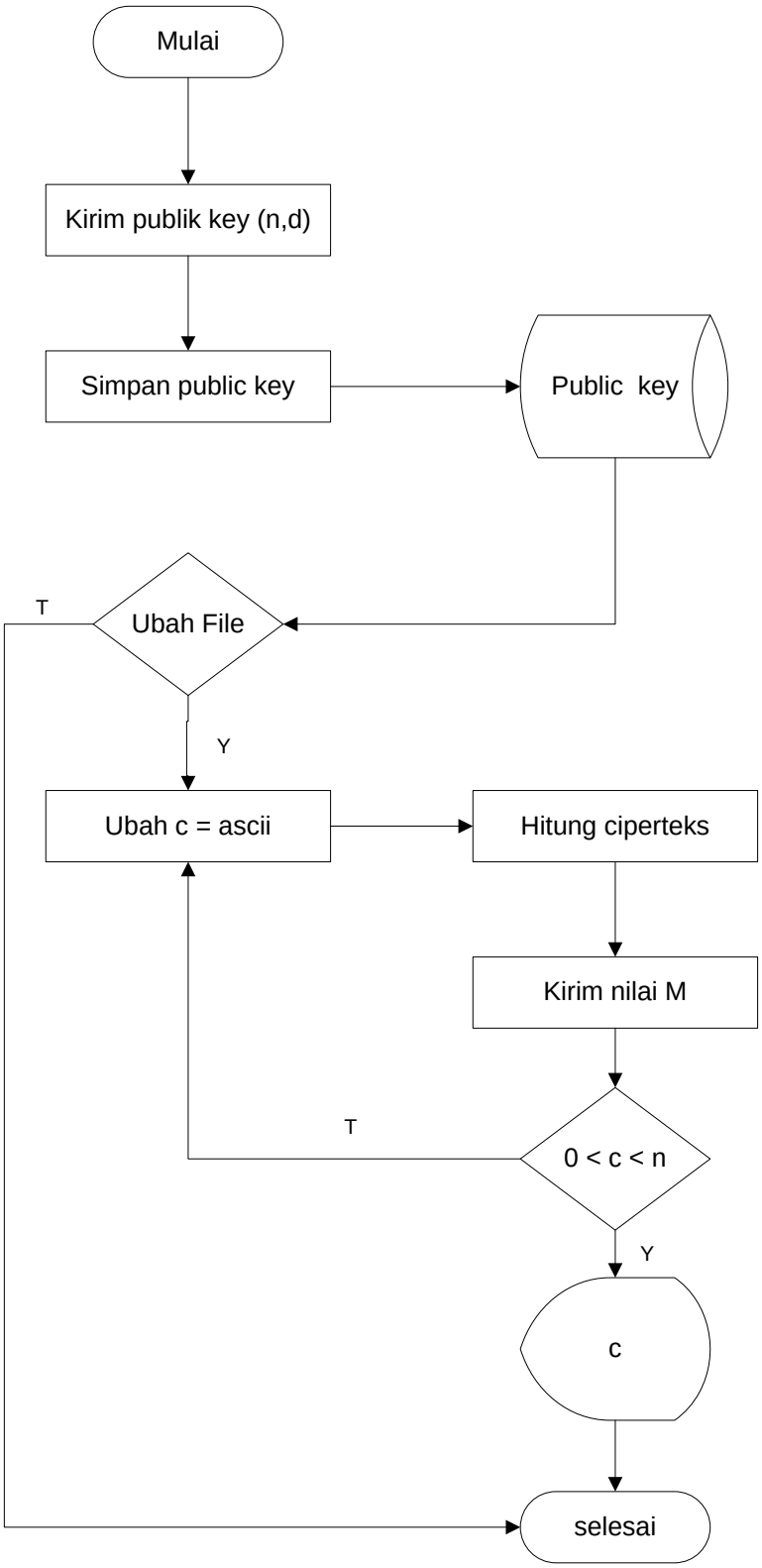
Gambar 3.20. *Flowchar*t Proses Enkripsi

3. Algoritma Proses Dekripsi

Operasi dekripsi tidak berbeda jauh dengan operasi *encrypt*, yang berbeda adalah nilai yang dimasukkan kedalam fungsi powmod itu. Dalam operasi *decrypt*, nilai M diganti dengan nilai c dari *ciphertext* (hasil enkripsi) dan nilai **e** dari *public* key diganti dengan nilai d dari *private* key, sedangkan nilai n dari *public* key selalu sama dengan nilai n dari *private* key.

Adapun langkah-langkah proses dekripsi dalam algoritma RSA adalah sebagai berikut :

a. Si A mengirimkan *private key* (n,d) nya untuk si B, dan menyimpan secara rahasia *public key*-nya.
b. Si B ingin mengubah *file* "c" pada si A.
c. Si B kemudian merubah “c” menjadi kode *ascii* (berupa *integer*).
d. Menghitung "M" (nilai yang telah terdekripsi) dengan menggunakan *public key* yang dikirimkan oleh si A kepadanya.
e. Kemudian B mengirimkan nilai “M” kepada A dengan menggunakan *private-key* miliknya.
f. Nilai c harus lebih besar dari 0, dan harus lebih kecil dari nilai n (dari *public key*).



Gambar 3.21. *Flowchar*t Proses Dekripsi

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dokumen adalah hal yang sangat penting bagi beberapa orang, termasuk para mahasiswa semester akhir yang telah berusaha menyelesaikan tugas akhirnya demi sebuah gelar sarjana dan untuk membuat orang tua bangga. Pada umumnya, dokumen tugas akhir tersebut akan dikumpulkan di kampus tempat mahasiswa tersebut belajar. Sebelum dibukukan, biasanya dokumen tersebut dijadikan *softcopy* di dalam kepingan CD yang diproses oleh pustakawan yang bertugas di perpustakaan kampus. Terlepas dari rasa kepercayaan terhadap petugas perpustakaan yang menjalankan tugasnya dengan baik, muncul rasa khawatir akan data – data yang ada di dalam dokumen tugas akhir mahasiswa disalah gunakan untuk kepentingan pribadi. Misalnya, menjual data yang ada kepada mahasiswa di kampus lain dengan bayaran yang cukup memuaskan hanya dengan menyalin data tersebut dan merubah sedikit kata atau kalimatnya tanpa sepengetahuan penulis dan tanpa sepengetahuan pihak kampus.

Hal – hal tersebut di atas besar kemungkinan terjadi jika dokumen tugas akhir mahasiswa tidak dilengkapi keamanan. Salah satu teknik untuk pengamanan data adalah dengan menggunakan algoritma penyandian data. Algoritma penyandian data saat ini semakin banyak jumlahnya, sejalan dengan berkembangnya ilmu yang mempelajari penyandian data tersebut. Ilmu ini biasa disebut Kriptografi. Metode yang cukup penting dalam pengamanan data, untuk

menjaga kerahasiaan suatu data dalam kriptografi salah satunya adalah enkripsi (*encryption*). Enkripsi adalah proses pengubahan pesan asli menjadi *ciphertext*. Sedangkan proses untuk mengubah pesan yang telah disembunyikan menjadi pesan asli disebut dekripsi. Pesan asli disebut *plaintext* sedangkan pesan yang sudah diubah atau disandikan supaya tidak mudah dibaca disebut dengan *ciphertext*.

Proses yang dilakukan dalam penelitian ini adalah dengan melakukan proses enkripsi dan dekripsi sebuah dokumen atau *file* yang disimpan atau dibuat menggunakan *Microsoft Office* dengan algoritma kriptografi Rhivest Shamir Adleman (RSA), dengan menggunakan alat bantu perancangan seperti *Unified Modelling Language* (UML) dan *Microsoft Visual Basic* 2010 untuk mempermudah dalam pembuatan aplikasi. Setelah perancangan selesai, *file* akan dienkripsi kemudian hasilnya adalah *file* tersebut berubah isi aslinya menjadi teks yang tidak dipahami maknanya. Dengan latar masalah di atas, penulis mengangkat judul **"Implementasi Keamanan Dokumen Office Dengan Algoritma Rhivest Shamir Adleman (RSA)"** untuk tugas akhir ini.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas dalam tugas akhir ini adalah:

1. Bagaimana penerapan algoritma kriptografi RSA untuk proses enkripsi dan dekripsi sebuah *file*?
2. Bagaimana *file* dapat dimasukkan ke aplikasi?
3. Bagaimana proses enkripsi dan dekripsi *file* yang ada?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Jenis data yang diolah adalah data yang dibuat atau disimpan menggunakan *Microsoft Office*.
2. Aplikasi ini bersifat *offline.*
3. Yang dienkripsi adalah dokumen.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Untuk menciptakan aplikasi keamanan data.
2. Untuk mencari tahu bagaimana proses enkripsi dan dekripsi dari Algoritma Kriptografi Rhivest Shamir Adleman (RSA).
3. Untuk menyelesaikan tugas akhir penulis.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Membuka pikiran pembaca bahwa dokumen apapun yang telah kita buat, hendaknya dijaga kemanannya termasuk skripsi atau tugas akhir.
2. Menambah pengetahuan pembaca tentang Algoritma Kriptografi Rhivest Shamir Adleman (RSA).
3. Membuat penulis untuk terus belajar dan mengembangkan ilmu yang telah didapat selama proses perkuliahan.

BAB II

Landasan Teori

2.1 Keamanan

Secara umum keamanan adalah proses untuk mencegah masuknya *user* yang tidak memiliki hak akses terhadap suatu sistem atau penyusup yang hadir di tengah – tengah lalu lintas data saat pengiriman sebuah *file* atau dokumen. Keamanan jaringan komputer sendiri bertujuan untuk mengantisipasi resiko pada jaringan komputer berupa bentuk ancaman fisik maupun *logic* baik langsung ataupun tidak langsung mengganggu aktivitas yang sedang berlangsung dalam jaringan komputer (M. Suyuti Ma'sum, dkk, 2017).

Keamanan informasi adalah suatu keharusan yang harus diperhatikan terutama jika informasi itu bersifar rahasia. Ketika suatu data dikirim melalui jaringan internet, data akan mengalami beberapa proses, proses tersebut terbagi oleh 7 layer yang tidak saling bergantungan antara satu sama lain tetapi saling berkaitan atau sering disebut 7 Layer OSI (*Open System Interconnection*). Model referensi OSI terdiri dari 7 buah bagian (*Layer*), yang masing – masing *layer* mempunyai tugas sendiri – sendiri, yang dibagi menjadi dua bagian yaitu proses *encapsulation* dan proses *decapsulation*, karena begitu kompleks proses dari pengiriman data, membuat proses tersebut tidak aman dari pihak ketiga, atau mengalami kerusakan pada proses pengirimannya (Sri wahyuni, dkk, 2018).

Aspek-aspek Keamanan Komputer. Keamanan komputer meliputi delapan aspek, antara lain:

a. *Authentication,* penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi. Dengan kata lain, informasi itu benar - benar datang dari orang yang dikehendaki.
b. *Integrity*, keaslian pesan yang dikirim melalui jaringan dan dapat dipastikan bahwa informasi yang dikirim tidak dimodifikasi orang yang tidak berhak.
c. *Non-repudiation,* merupakan hal yang berhubungan dengan si pengirim. Pengirim tidak dapat mengelak bahwa dialah yang mengirim informasi tersebut.
d. *Authority*, informasi yang berada pada sistem jaringan tidak dapat dimodifikasi oleh pihak yang tidak berhak untuk mengaksesnya.
e. *Confidentiality*, merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Kerahasiaan ini biasanya berhubungan dengan informasi yang diberikan ke pihak lain.
f. *Privacy*, lebih ke arah data-data yang bersifat pribadi.
g. *Availability*, aspek availabilitas berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.
h. *Access Control*, aspek ini berhubungan dengan cara pengaturan akses ke informasi. Hal ini biasanya berhubungan dengan masalah

otentikasi dan privasi. Kontrol akses seringkali dilakukan dengan menggunakan kombinasi *user id* dan *password* ataupun dengan mekanisme lain.

2.2 Kriptografi

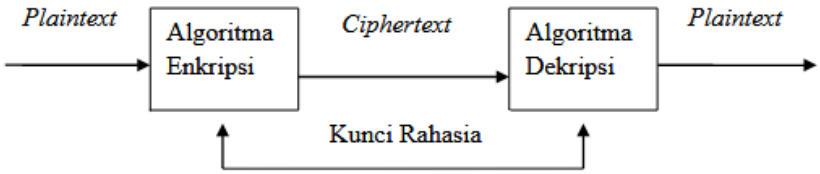
2.2.1 Definisi Kriptografi

Kriptografi (*cryptography*) berasal dari Bahasa Yunani yaitu "*cryptos*" yang artinya "*secret*" (rahasia) dan "*graphein*" yang artinya "*writing*" (menulis). Jadi kriptografi berarti "*secret writing*" (tulisan rahasia). Kriptografi pada awalnya dijabarkan sebagai ilmu yang memepelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data, dan otentikasi entitas (Sadikin, Rifki, 2018).

2.2.2 Jenis Algoritma Kriptografi

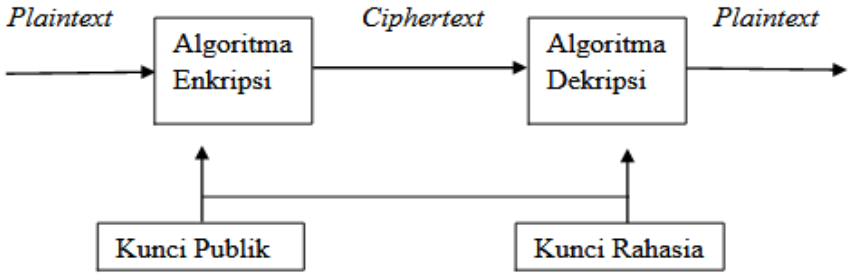
Terdapat 3 algoritma pada kriptografi modern (Rachman, 2010), yaitu:

1. Algoritma simetris adalah algoritma yang menggunakan kunci yang sama untuk enkripsi dan dekripsinya. Aplikasinya digunakan oleh algoritma *Data Encryption Standard (DES)*, *Advance Encryption Standard (AES)*, *International Data Encryption Algoritma (IDEA)*, A5, RC4. Skema kriptografi simetris dapat dilihat pada gambar 2.1



Gambar 2.1 Skema kriptografi simetris

2. Algoritma Asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan satu lagi untuk dekripsi. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA. Skema kriptografi asimetris dapat dilihat pada gambar 2.2



Gambar 2.2 Skema kriptografi asimetris

3. Algoritma hibrida adalah algoritma yang memanfaatkan dua tingkatan kunci, yaitu kunci rahasia (simetri) yang disebut juga session key (kunci sesi) untuk enkripsi data dan pasangan kunci rahasia – kunci publik untuk pemberian tanda tangan digital serta melindungi kunci simetri. Algoritma kriptografi yang beroperasi dalam mode bit dapat dikelompokkan menjadi dua kategori: Cipher aliran (*stream cipher*) dan Cipher blok (*block cipher*) (Rachman, 2010).

2.2.3 Kekuatan Algoritma Kriptografi

Dari ketiga algoritma diatas Algoritma kriptografi harus memiliki kekuatan untuk melakukan (Rivalri, dkk, 2014):

1. Konfusi / pembinggungan (*confusion*), dari teks terang sehingga sulit untuk direkrontruksikan secara langsung tanpa menggunakan algoritma dekripsinya.
2. Difusi / peleburan (*diffusion*), dari teks terang sehingga karakteristik dari teks terang tersebut hilang sehingga dapat digunakan untuk mengamankan informasi.

2.2.4 Tujuan Kriptografi

Tujuan kriptografi adalah (Safrina, Nanda, 2017):

1. Kerahasiaan (*confidentiality*)

   Kerahasiaan adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak - pihak yang tidak berhak. Di dalam kriptografi, layanan ini direalisasikan dengan menyandikan pesan menjadi cipherteks.
2. Integritas (*integrity*)

   Integritas adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi pesan oleh pihak-pihak yang tidak berhak, antara lain

penyisipan, pengahapusan, dan pensubtitusian data lain kedalam pesan yang sebenarnya.

3. Otentikasi (*authentication*)

   Otentikasi adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak – pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan. Dua pihak yang saling berkomunikasi harus dapat mengotentikasi satu sama lain sehingga ia dapat memastikan sumber pesan.

4. Nirpenyangkalan (*Non-repidiation*)

   Nirpenyangkalan adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan memberi otoritas kepada penerima pesan untuk melakuka pembelian, namun kemudian ia menyangkal telah melakukan pembeli.

2.2.5 Istilah Dalam Kriptografi

Ada beberapa istilah dalam kriptografi yang sering dijumpai, berikut adalah istilah – istilah dan pengertiannya:

1. *Plaintext* dan *Ciphertext*

   Plaintext adalah teks asli atau pesan asli yang dapat dimengerti, dibaca dengan jelas dan dapat dipahami maknanya. Sedangkan *Ciphertext* adalah pesan atau teks asli yang sudah tersandi sehingga sudah menjadi kode – kode yang sulit dipahami artinya.

2. Pengirim dan Penerima

   Pengirim adalah *user* yang mengirim pesan kepada *user* lainnya. Pengirimlah yang akan mengirim *file* beserta kunci untuk membuka data yang telah dienkripsi. Penerima adalah *user* yang menerima data atau informasi yang memiliki hak akses terhadap *file* yang diberikan dan mengetahui kunci untuk mengembalikan teks ke dalam bentuk semula (*plaintext*).

3. Enkripsi dan Dekripsi

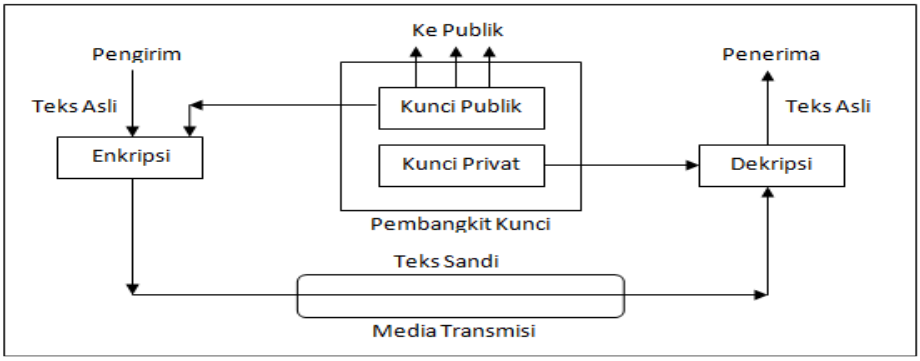
   Enkripsi adalah proses penyandian, teks asli (*plaintext*) diubah menjadi *ciphertext* (teks tersandi). Dekripsi adalah proses pengembalian teks ke dalam bentuk awal.

2.3 RSA (Rhivest Shamir Adleman)

RSA merupakan algoritma kriptografi kunci publik atau sering disebut kunci asimetrik (kunci enkripsi dan kunci dekripsi berbeda) sehingga tidak membutuhkan saluran yang aman untuk distribusi kunci (Arief, dkk, 2016). RSA ditemukan oleh tiga peneliti dari MIT (Massachussets Institute of Technology), yaitu Ronald Linn Rivest, Adi Shamir, dan Len Adleman pada tahun 1977.

Algoritma enkripsi dan dekripsi sistem kriptografi RSA bersandar pada asumsi fungsi satu arah (*one-way function*) yang dibangun oleh fungsi eksponensial modular pada grup perkalian $\left(Z_{n,}^{*} \times\right)$ dan grup perkalian $(Z_{\emptyset(n),}^{*} \times)$

dengan $n = p \times q,\ p, q$ adalah bilangan prima dan $\emptyset(n) = (p-1)(q-1)$ (Sadikin, rifki, 2018).



Gambar 2.3 Sistem Kriptografi dengan kunci publik RSA

2.3.1 Algoritma – algoritma Sistem Kriptografi RSA

Terdapat 3 algoritma pada sistem kriptografi RSA, yaitu algoritma pembangkit kunci, algoritma enkripsi, dan algoritma dekripsi. (Sadikin, rifki, 2018):

1. Pembangkit Kunci RSA

   Untuk membangun kunci RSA terlebih dahulu pendekripsi membangkitkan sepasang kunci yaitu kunci *public* dan kunci *privat*. Langkah – langkah untuk membangkitkan kunci:

   a. Pilih 2 bilangan prima besar untuk nilai p dan q
   b. Hitung nilai modulus n = p x q
   c. Hitung menggunakan fungsi Euler ∅(n) = (p-1) x (q-1)
   d. Pilih nilai integer e acak sebagai kunci publik, dengan syarat memenuhi Greater Common Divisor (GCD) (e, ∅(n)) = 1, 1 < e < ∅(n) (3)
   e. Hitung kunci privat d sehingga d x e = 1 (mod ∅(n)) (4)

2. Enkripsi

   Setelah kunci *public* K_{public} dibangkitkan olek pendekripsi maka sembarang orang dapat menggunakan kunci *public* untuk mengirim teks sandi (Sadikin, rifki, 2018).

Setelah kunci publik telah dibangkitkan, maka proses enkripsi sudah bisa dilakukan dengan cara berikut ini:

a. Menggunakan kunci publik penerima pesan, *e*, dan modulus *n*.
b. Representasikan pesan atau *plaintext* kedalam kode ASCII sebelum dilakukan proses enkripsi selanjutnya.
c. Untuk menyandikan sebuah pesan *m* dengan menggunakan kunci publik *e*, kita menggunakan operasi :

$$c_i \equiv m_i^e \bmod n$$

d. Nyatakan *plaintext* menjadi blok-blok $m_1, m_2, \ldots, m_r$ sedemikian hingga setiap blok merepresentasikan nilai di dalam selang [0, n – 1].
e. Setiap blok m_i dienkripsikan menjadi blok c_i

3. Dekripsi

   Penerima yang mendapat teks sandi yang dienkripsi dengan kunci *public* maka penerima dapat menggunakan kunci privatnya untuk mengembalikan dalam bentuk teks asli. Tahapannya adalah sebagai berikut:

   a. Menggunakan kunci privat pesan *d*, modulus *n*. untuk menghasilkan m atau pesan *plaintext*.

b. Pesan (*ciphertext*) dibagi per-blok seperti pada saat proses enkripsi pesan dan dikembalikan menjadi pesan *plaintext* dengan menggunakan operasi sebagai berikut :

$$m_i \equiv c_i^d \bmod n$$

c. Setelah dilakukan proses dekripsi maka sudah didapatkan *plaintext*.

2.3.2 Keamanan RSA

RSA merupakan algoritma kriptografi yang cukup aman jika memilih paramater kunci dan implementasi yang tepat. Ancaman terjadi jika paramater dan implementasi lemah sehingga rentan untuk diserang. Perlu diketahui ancaman – ancaman terhadap RSA untuk membuat implementasi RSA lebih aman.

Faktorisasi adala tipe salah satu tipe penyerangan yang bisa melemahkan keamanan RSA (Sadikin, rifki, 2018) perhatikan pada kunci publik RSA terdapat parameter n yang merupakan bilangan komposit hasil perkalian p dan q. Padahal p dan q merupakan parameter privat (harus dirahasiakan) jadi apabila penyerang (Eve) mampu memfaktorisasikan n sehingga menemukan p dan q sistem kriptigrafi RSA menjadi kehilangan keamanannya. Jika p dan q diketahui maka Eve dapat menghitung $\emptyset(n) = (p-1)(q-1)$ dan dapat mengetahui pasangan eksponen e pada kunci publik yaitu $d = e^{-1}$ pada $Z_{\emptyset}^{*}(n)$ sehingga Eve dapat mendekripsi teks sandi tanpa perlu tahu kunci privat (Sadikin, rifki, 2018).

Selain ukuran bit parameter n ada beberapa hal yang perlu diperhatikan ketika membangkitkan parameter RSA untuk bertahan dari serangan faktorisasi. Rekomendasi nilai n, p dan q disrankan memenuhi hal – hal berikut (Sadikin, rifki, 2018):

a. Ukuran $n \geq 1024$ bit.
b. Selisih panjang bit p dan q harus kecil dan setidaknya berukura 512 bit.
c. $(p-1)$ dan $(q-1)$ memiliki faktor bilangan prima besar.\
d. gcd$(p-1, q-1)$ harus bilangan integer kecil.
e. Untuk menghindari faktorisasi bisa dipilih terlebih dahulu p_1 dan q_1 yang merupakan bilangan prima. Lalu hitung $p = 2_{p1} + 1$ dan $q = 2_{q1} + 1$ jika p dan q adalah bilangan prima maka terima p dan q.

Berikut ini jenis serangan lain terhadap kunci publik RSA:

1. Serangan terhadap eksponen yang dipakai untuk enkripsi (e) Jika e terlalu kecil menurutu teorema Coppersmith maka terdapat algoritma dengan kompleksitas $\log(n)$ yang dapat menemukan $\frac{1}{e}$ sehingga dapat membuka teks asli dari teks sandi $C = P^e \, mod \, n$. Untuk menghindari serangan jenis ini direkomendasikan nilai $e \approx 2^{16} + 1$
2. Serangan *Chosen-Ciphertext* Jika Eve mampu menyadap C dan Penerima selalu mendekripsi sembarang teks sandi dari Eve dengan nilai n yang sama, maka Eve dapat mengetahui teks asli P atas C dengan algoritma *Extended Euclid.*
3. Serangan terhadap eksponen dekripsi. Ekspnen dekripsi yang bernilai rendah bisa menjadi sumber kelemahan sistem kriptografi RSA. Direkomendasikan nilai $d \geq \frac{1}{3} n^{1/4}$ agar aman dari serangan terhadap eksponen dekripsi yang rendah.

4. Serangan dengan menggunakan teks asli. Teks asli dan teks sandi pada sistem kriptografi RSA merupakan bilangan integer antara 0 sampai dengan $n-1$. Penyerangan teks asli ini memanfaatkan informasi ini. Salah satu serangan yang dapat membahayakan sistem kriptografi RSA adalah penggunaan teks asli dengan bilangan integer yang kecil. Serangan ini disebut *Short Message Attack.* Agar sistem kriptografi RSA terhindar dari serangan *Short Message Attack* perlu digunakan algoritma *padding* yang khusus. Kelompok RSA yang merekomendasikan penggunaan algoritma *padding* yang disebut dengan *Optimal Asymmetric Encryption Padding* (Sadikin, rifki, 2018).

Penggunaan algoritma RSA (Rhivest Shamir Adlemant) menggunakan kode ASCII (*Standard Code for Information Interchange*), berikut ini adalah tabel ASCII .

DEC	***OCT***	***HEX***	***BIN***	***CHAR***	**Keterangan**	**DEC**	**OCT**	**HEX**	**BIN**	**CHAR**	**Keterangan**
0	0	0	0	NUL	*Null* (tidak terlihat)	128	200	80	10000000	€	*Euro sign*
1	1	1	1	SOH	*Start of heading* (tidak terlihat)	129	201	81	10000001	•	
2	2	2	10	STX	*Start of text* (tidak terlihat)	130	202	82	10000010	‚	*Single low-9 quotation mark*
3	3	3	11	ETX	*End of text* (tidak terlihat)	131	203	83	10000011	*ƒ*	*Latin small letter f with hook*
4	4	4	100	EOT	*End of transmission* (tidak terlihat)	132	204	84	10000100	„	*Double low-9 quotation mark*
5	5	5	101	ENQ	*Enquiry* (tidak terlihat)	133	205	85	10000101	…	*Horizontal ellipsis*
6	6	6	110	ACK	*Acknowledge* (tidak terlihat)	134	206	86	10000110	†	*Dagger*

7	7	7	111	BEL	*Bell* (tidak terlihat)	135	207	87	10000111	‡	*Double dagger*
8	10	8	1000	BS	*Backspace*	136	210	88	10001000	ˆ	*Modifier letter circumflex accent*
9	11	9	1001	HT	*Horizontal tabulation*	137	211	89	10001001	‰	*Per mille sign*
10	12	A	1010	LF	Pergantian baris *(Line feed)*	138	212	8A	10001010	Š	*Latin capital letter S with caron*
11	13	B	1011	VT	Tabulasi vertikal	139	213	8B	10001011	‹	*Single left-pointing angle quotation*
12	14	C	1100	FF	Pergantian baris *(Form feed)*	140	214	8C	10001100	Œ	*Latin capital ligature OE*
13	15	D	1101	CR	Pergantian baris *(carriage return)*	141	215	8D	10001101	•	
14	16	E	1110	SO	*Shift out* (tidak terlihat)	142	216	8E	10001110	Ž	*Latin captial letter Z with caron*
15	17	F	1111	SI	*Shift in* (tidak terlihat)	143	217	8F	10001111	•	
16	20	10	10000	DLE	*Data link escape* (tidak terlihat)	144	220	90	10010000	•	
17	21	11	10001	DC1	*Device control* 1 (tidak terlihat)	145	221	91	10010001	‘	*Left single quotation mark*
18	22	12	10010	DC2	*Device control* 2 (tidak terlihat)	146	222	92	10010010	’	*Right single quotation mark*
19	23	13	10011	DC3	*Device control* 3 (tidak terlihat)	147	223	93	10010011	“	*Left double quotation mark*
20	24	14	10100	DC4	Device control 4 (tidak terlihat)	148	224	94	10010100	”	*Right double quotation mark*
21	25	15	10101	NAK	*Negative acknowledge* (tidak terlihat)	149	225	95	10010101	•	*Bullet*
22	26	16	10110	SYN	*Synchronous idle* (tidak terlihat)	150	226	96	10010110	–	*En dash*
23	27	17	10111	ETB	*End of transmission block* (tidak terlih	151	227	97	10010111	—	*Em dash*

					at)						
24	30	18	11000	CAN	*Cancel* (tidak terlih	152	230	98	10011000	˜	*Small tilde*
25	31	19	11001	EM	*End of medium* (tidak terlih at)	153	231	99	10011001	™	*Trade mark sign*
26	32	1A	11010	SUB	*Substitute* (tidak terlih at)	154	232	9A	10011010	š	*Latin small letter S with caron*
27	33	1B	11011	ESC	*Escape* (tidak terlih at)	155	233	9B	10011011	›	*Single right-pointing angle quotation mark*
28	34	1C	11100	FS	*File* separator	156	234	9C	10011100	œ	Latin *small ligature* oe
29	35	1D	11101	GS	*Group separator*	157	235	9D	10011101	•	
30	36	1E	11110	RS	*Record separator*	158	236	9E	10011110	ž	Latin *small letter z with caron*
31	37	1F	11111	US	*Unit separator*	159	237	9F	10011111	Ÿ	Latin *capital letter* Y *with diaeresis*
32	40	20	100000	spasi	Spasi	160	240	A0	10100000		Spasi yang bukan pemisah kata
33	41	21	100001	!	Tanda seru *(exclamation)*	161	241	A1	10100001	¡	Tanda seru terbalik
34	42	22	100010	"	Tanda kutip dua	162	242	A2	10100010	¢	Tanda sen *(Cent)*
35	43	23	100011	#	Tanda pagar (kres)	163	243	A3	10100011	£	Tanda *Poundsterling*
36	44	24	100100	$	Tanda mata uang dolar	164	244	A4	10100100	¤	Tanda mata uang *(Currency)*
37	45	25	100101	%	Tanda persen	165	245	A5	10100101	¥	Tanda *Yen*
38	46	26	100110	&	Karakter *ampersand* (&)	166	246	A6	10100110	¦	Garis tegak putus-putus
39	47	27	100111	'	Karakter *Apostrof*	167	247	A7	10100111	§	*Section sign*
40	50	28	101000	(	Tanda kurung buka	168	250	A8	10101000	¨	*Spacing diaeresis - umlaut*
41	51	29	101001	)	Tanda kurung tutup	169	251	A9	10101001	©	Tanda hak cipta *(Copyright)*
42	52	2A	101010	*	Karakter asterisk (bintang)	170	252	AA	10101010	ª	*Feminine ordinal indicator*

43	53	2B	101011	+	Tanda tambah (plus)	171	253	AB	10101011	«	*Left double angle quotes*
44	54	2C	101100	,	Karakter koma	172	254	AC	10101100	¬	*Not sign*
45	55	2D	101101	-	Karakter hyphen (strip)	173	255	AD	10101101		Tanda *strip (hyphen)*
46	56	2E	101110	.	Tanda titik	174	256	AE	10101110	®	Tanda merk terdaftar
47	57	2F	101111	/	Garis miring (slash)	175	257	AF	10101111	¯	*Spacing Macron (Macron)*
48	60	30	110000	0	Angka nol	176	260	B0	10110000	°	Tanda derajat
49	61	31	110001	1	Angka satu	177	261	B1	10110001	±	Tanda kurang lebih (plus-minus)
50	62	32	110010	2	Angka dua	178	262	B2	10110010	²	Tanda kuadrat (pangkat dua)
51	63	33	110011	3	Angka tiga	179	263	B3	10110011	³	Tanda kubik (pangkat tiga)
52	64	34	110100	4	Angka empat	180	264	B4	10110100	´	*Acute accent*
53	65	35	110101	5	Angka lima	181	265	B5	10110101	µ	*Micro sign*
54	66	36	110110	6	Angka enam	182	266	B6	10110110	¶	*Pilcrow sign*
55	67	37	110111	7	Angka tujuh	183	267	B7	10110111	·	*Middle dot*
56	70	38	111000	8	Angka delapan	184	270	B8	10111000	¸	*Spacing cedilla*
57	71	39	111001	9	Angka sembilan	185	271	B9	10111001	¹	*Superscript one*
58	72	3A	111010	:	Tanda titik dua	186	272	BA	10111010	º	*Masculine ordinal indicator*
59	73	3B	111011	;	Tanda titik koma	187	273	BB	10111011	»	*Right double angle quotes*
60	74	3C	111100	<	Tanda lebih kecil	188	274	BC	10111100	¼	*Fraction one quarter*
61	75	3D	111101	=	Tanda sama dengan	189	275	BD	10111101	½	*Fraction one half*
62	76	3E	111110	>	Tanda lebih besar	190	276	BE	10111110	¾	*Fraction three quarters*
63	77	3F	111111	?	Tanda tanya	191	277	BF	10111111	¿	*Inverted question mark*

64	100	40	1000000	@	A keong (@)	192	300	C0	11000000	À	Latin *capital letter* A *with grave*
65	101	41	1000001	A	Huruf latin A kapital	193	301	C1	11000001	Á	Latin *capital letter* A *with acute*
66	102	42	1000010	B	Huruf latin B kapital	194	302	C2	11000010	Â	Latin *capital letter* A *with circumflex*
67	103	43	1000011	C	Huruf latin C kapital	195	303	C3	11000011	Ã	Latin *capital letter* A *with tilde*
68	104	44	1000100	D	Huruf latin D kapital	196	304	C4	11000100	Ä	Latin *capital letter* A *with diaeresis*
69	105	45	1000101	E	Huruf latin E kapital	197	305	C5	11000101	Å	Latin *capital letter* A *with ring above*
70	106	46	1000110	F	Huruf latin F kapital	198	306	C6	11000110	Æ	Latin *capital letter* AE
71	107	47	1000111	G	Huruf latin G kapital	199	307	C7	11000111	Ç	Latin *capital letter* C *with cedilla*
72	110	48	1001000	H	Huruf latin H kapital	200	310	C8	11001000	È	Latin *capital letter* E *with grave*
73	111	49	1001001	I	Huruf latin I kapital	201	311	C9	11001001	É	Latin *capital letter* E *with acute*
74	112	4A	1001010	J	Huruf latin J kapital	202	312	CA	11001010	Ê	Latin *capital letter* E *with circumflex*
75	113	4B	1001011	K	Huruf latin K kapital	203	313	CC	11001011	Ë	Latin *capital letter* E *with diaeresis*
76	114	4C	1001100	L	Huruf latin L kapital	204	314	CB	11001100	Ì	Latin *capital letter* I *with grave*
77	115	4D	1001101	M	Huruf latin M kapital	205	315	CD	11001101	Í	Latin *capital letter* I *with acute*
78	116	4E	1001110	N	Huruf latin N kapital	206	316	CE	11001110	Î	Latin *capital letter* I *with circumflex*
79	117	4F	1001111	O	Huruf latin O kapital	207	317	CF	11001111	Ï	Latin *capital letter* I *with diaeresis*

80	120	50	1010000	P	Huruf latin P kapital	208	320	D0	11010000	Đ	Latin *capital letter ETH*
81	121	51	1010001	Q	Huruf latin Q kapital	209	321	D1	11010001	Ñ	Latin *capital letter N with tilde*
82	122	52	1010010	R	Huruf latin R kapital	210	322	D2	11010010	Ò	*Latin capital letter O with grave*
83	123	53	1010011	S	Huruf latin S kapital	211	323	D3	11010011	Ó	*Latin capital letter O with acute*
84	124	54	1010100	T	Huruf latin T kapital	212	324	D4	11010100	Ô	Latin *capital letter* O *with circumflex*
85	125	55	1010101	U	Huruf latin U kapital	213	325	D5	11010101	Õ	Latin *capital letter* O *with tilde*
86	126	56	1010110	V	Huruf latin V kapital	214	326	D6	11010110	Ö	Latin *capital letter* O *with diaeresis*
87	127	57	1010111	W	Huruf latin W kapital	215	327	D7	11010111	×	*Multiplication sign*
88	130	58	1011000	X	Huruf latin X kapital	216	330	D8	11011000	Ø	Latin *capital letter* O *with slash*
89	131	59	1011001	Y	Huruf latin Y kapital	217	331	D9	11011001	Ù	Latin *capital letter* U *with grave*
90	132	5A	1011010	Z	Huruf latin Z kapital	218	332	DA	11011010	Ú	Latin *capital letter* U *with acute*
91	133	5B	1011011	[	Kurung siku kiri	219	333	DB	11011011	Û	Latin *capital letter* U *with circumflex*
92	134	5C	1011100	\	Garis miring terbalik (*backslash*)	220	334	DC	11011100	Ü	Latin *capital letter* U *with diaeresis*
93	135	5D	1011101	]	Kurung sikur kanan	221	335	DD	11011101	Ý	Latin *capital letter* Y *with acute*
94	136	5E	1011110	^	Tanda pangkat	222	336	DE	11011110	Þ	Latin *capital letter THORN*
95	137	5F	1011111	_	Garis bawah *(underscore)*	223	337	DF	11011111	ß	Latin *small letter sharp s - ess-zed*
96	140	60	1100000	`	Tanda petik satu	224	340	E0	11100000	à	Latin *small letter* a *with*

											grave
97	141	61	1100001	a	Huruf latin a kecil	225	341	E1	11100001	á	Latin *small letter* a *with acute*
98	142	62	1100010	b	Huruf latin b kecil	226	342	E2	11100010	â	Latin *small letter* a *with circumflex*
99	143	63	1100011	c	Huruf latin c kecil	227	343	E3	11100011	ã	Latin *small letter* a *with tilde*
100	144	64	1100100	d	Huruf latin d kecil	228	344	E4	11100100	ä	Latin *small letter* a *with diaeresis*
101	145	65	1100101	e	Huruf latin e kecil	229	345	E5	11100101	å	Latin *small letter* a *with ring above*
102	146	66	1100110	f	Huruf latin f kecil	230	346	E6	11100110	æ	Latin *small letter* ae
103	147	67	1100111	g	Huruf latin g kecil	231	347	E7	11100111	ç	Latin *small letter* c *with cedilla*
104	150	68	1101000	h	Huruf latin h kecil	232	350	E8	11101000	è	Latin *small letter* e *with grave*
105	151	69	1101001	i	Huruf latin i kecil	233	351	E9	11101001	é	Latin *small letter* e *with acute*
106	152	6A	1101010	j	Huruf latin j kecil	234	352	EA	11101010	ê	Latin *small letter* e *with circumflex*
107	153	6B	1101011	k	Huruf latin k kecil	235	353	EB	11101011	ë	Latin *small letter* e *with diaeresis*
108	154	6C	1101100	l	Huruf latin l kecil	236	354	EC	11101100	ì	Latin *small letter* i *with grave*
109	155	6D	1101101	m	Huruf latin m kecil	237	355	ED	11101101	í	Latin *small letter* i *with acute*
110	156	6E	1101110	n	Huruf latin n kecil	238	356	EE	11101110	î	Latin *small letter* i *with circumflex*
111	157	6F	1101111	o	Huruf latin o kecil	239	357	EF	11101111	ï	Latin *small letter* i *with diaeresis*
112	160	70	1110000	p	Huruf latin p kecil	240	360	F0	11110000	ð	Latin *small letter eth*
113	161	71	1110001	q	Huruf latin q kecil	241	361	F1	11110001	ñ	Latin *small letter* n *with tilde*
114	162	72	1110010	r	Huruf latin r kecil	242	362	F2	11110010	ò	Latin *small letter* o *with grave*
115	163	73	1110011	s	Huruf latin s kecil	243	363	F3	11110011	ó	Latin *small letter* o *with acute*
116	164	74	1110100	t	Huruf latin t kecil	244	364	F4	11110100	ô	Latin *small letter* o *with circumflex*

117	165	75	1110101	u	Huruf latin u kecil	245	365	F5	11110101	õ	Latin *small letter* o *with tilde*
118	166	76	1110110	v	Huruf latin v kecil	246	366	F6	11110110	ö	Latin *small letter* o *with diaeresis*
119	167	77	1110111	w	Huruf latin w kecil	247	367	F7	11110111	÷	*Division sign*
120	170	78	1111000	x	Huruf latin x kecil	248	370	F8	11111000	ø	Latin *small letter* o *with slash*
121	171	79	1111001	y	Huruf latin y kecil	249	371	F9	11111001	ù	Latin *small letter* u *with grave*
122	172	7A	1111010	z	Huruf latin z kecil	250	372	FA	11111010	ú	Latin *small letter* u *with acute*
123	173	7B	1111011	{	Kurung kurawal buka	251	373	FB	11111011	û	Latin *small letter* u *with circumflex*
124	174	7C	1111100	|	Garis vertikal (pipa)	252	374	FC	11111100	ü	Latin *small letter* u *with diaeresis*
125	175	7D	1111101	}	Kurung kurawal tutup	253	375	FD	11111101	ý	Latin *small letter y with acute*
126	176	7E	1111110	~	Karakter gelombang *(tilde)*	254	376	FE	11111110	þ	Latin *small letter thorn*
127	177	7F	1111111	DEL	*Delete*	255	377	FF	11111111	ÿ	Latin *small letter* y *with diaeresis*

Tabel 2.1 ASCII

BAB IV

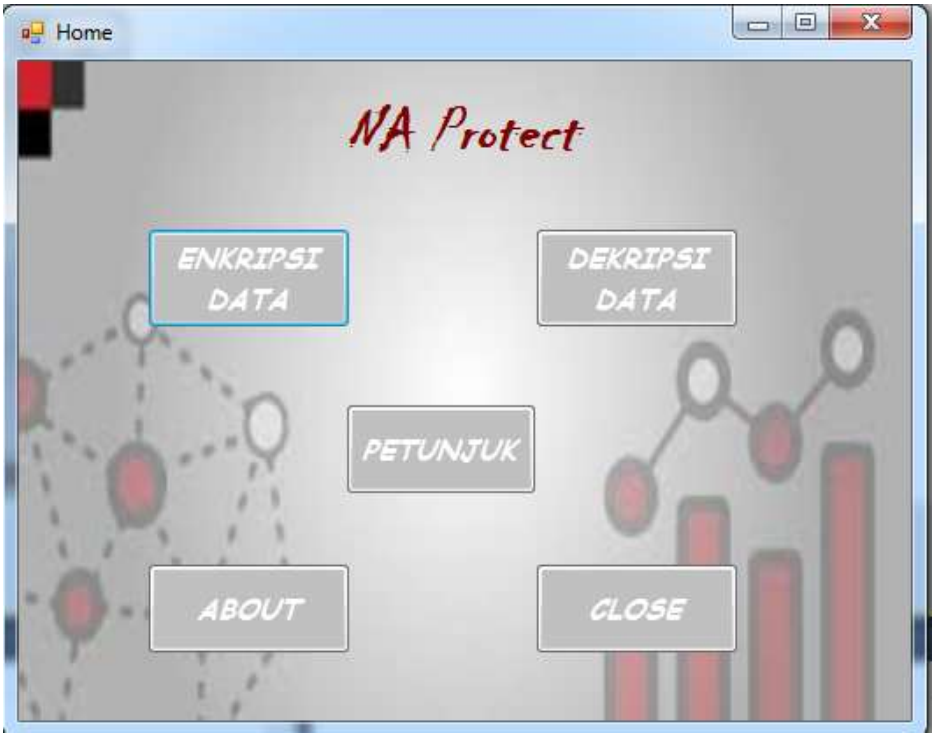
Hasil dan Pembahasan

4.1 Kebutuhan Spesifikasi Minimum Hardware dan Software

1. Spesifikasi komputer standard *Processor Pentium IV* 2,6 GHz, memori 512 MB, kartu grafik 128 MB.
2. Dapat digunakan minimal pada sistem operasi Microsoft Windows XP/Vista/7 secara *stand alone*.

4.2 Pengujian Aplikasi dan Pembahasan

1. Tampilan Awal Program



Gambar 4.2. Tampilan Awal

Di tampilan ini ada beberapa menu, diantaranya:

a. Enkripsi Data: Untuk menuju proses pengekripsian data

b. Dekripsi Data: Untuk menuju proses pendekripsian data

c. Petunjuk : Menampilkan petunjuk cara penggunaan

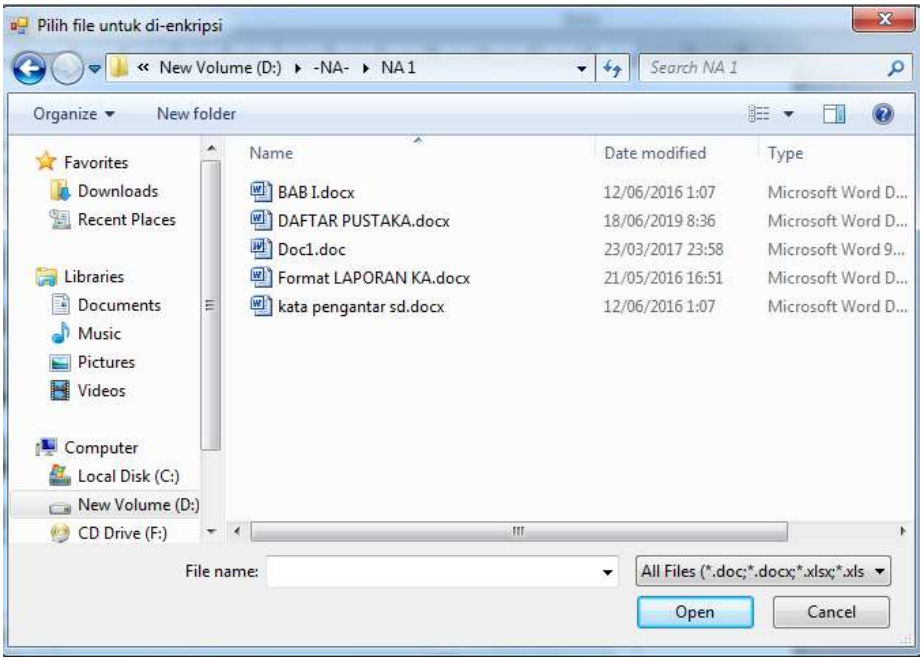
d. *About* : Menampilkan identitas orang yang membuat program

e. *Close* : Menutup Program

2. Tampilan *form* enkripsi data

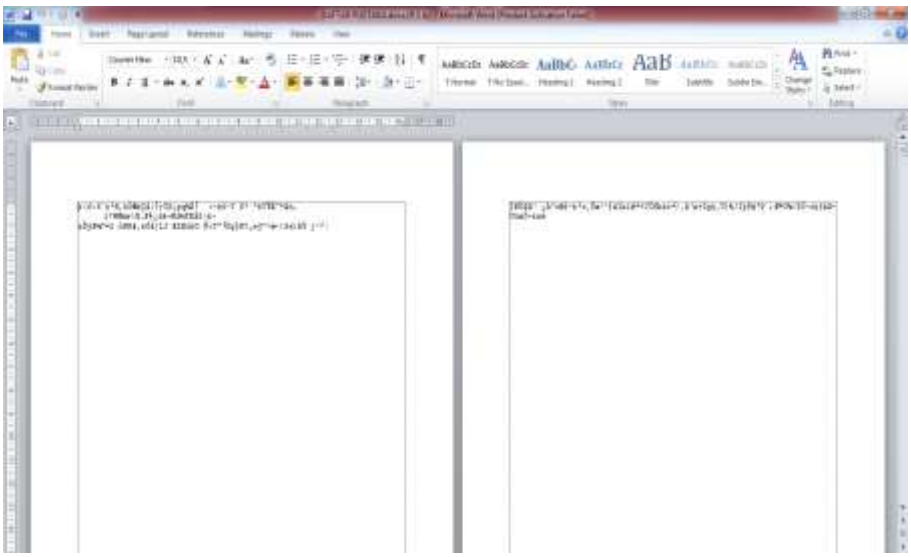
Gambar 4.3. Tampilan Enkripsi Data

Pada *form* ini, *user* bisa langsung memasukkan *file* yang akan dienkripsi dengan meng*klik* pada tombol *"browse"*. Setelah sudah memilih *file*, *user* bisa menyimpan hasil enkripsi di tempat yang berbeda dari asalnya dengan meng*klik* tombol "*save"* dan pilih folder yang diinginkan untuk menyimpan hasilnya. Setelah itu masukkan kunci ke dalam kolom kunci. Klik tombol "Enkripsi" dan *file* akan diproses.



Gambar 4.4. Tampilan *Browsing File*

Gambar 4.5. Tampilan proses enkripsi



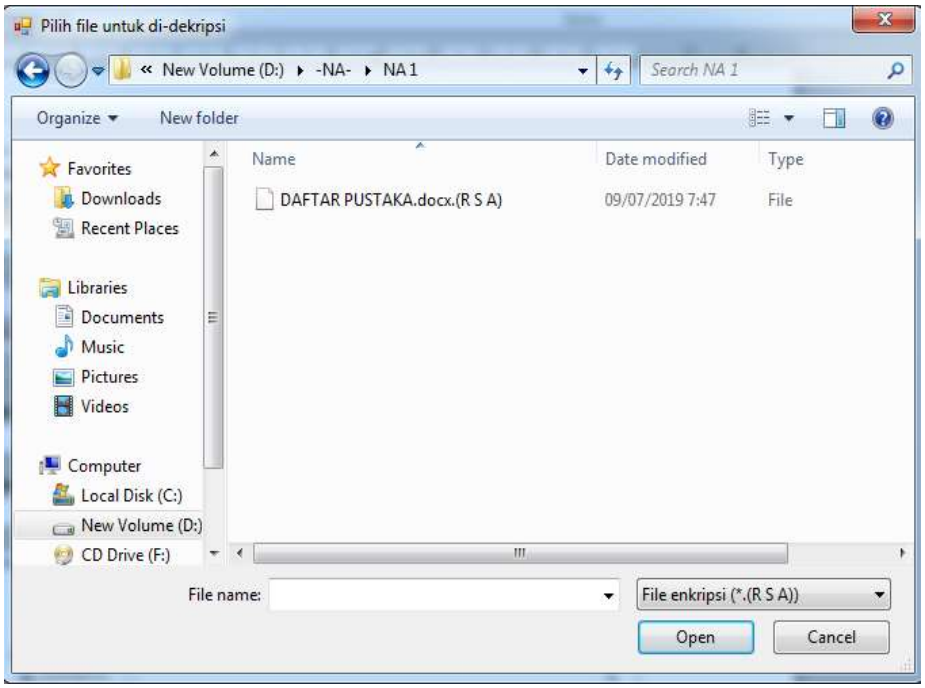
Gambar 4.6. Tampilan *file* yang telah dienkripsi

3. Tampilan *form* dekripsi data

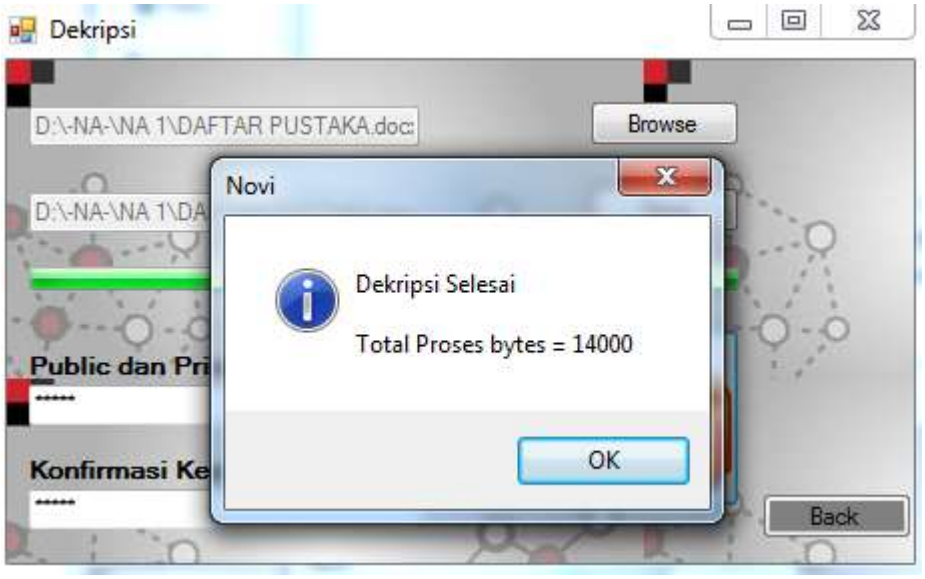
Gambar 4.7. Tampilan dekripsi data

Pada form ini, user bisa langsung memasukkan file yang akan dikembalikan ke bentuk awal, dengan mengklik pada tombol “browse”. Setelah sudah memilih file, user bisa menyimpan hasil dekripsi di tempat yang berbeda dari asalnya dengan mengklik tombol “save” dan pilih folder yang diinginkan untuk menyimpan hasilnya. Setelah itu masukkan kunci ke dalam kolom kunci,

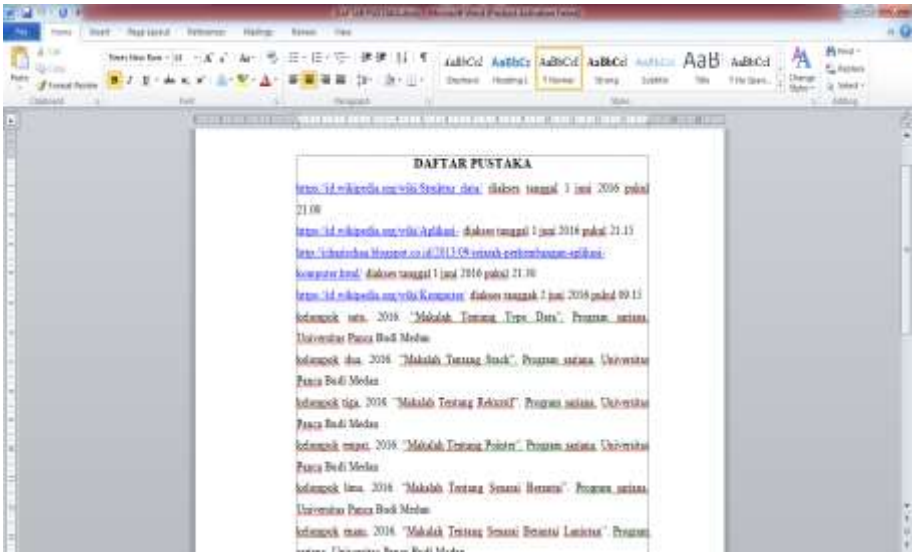
kunci untuk dekripsi harus sama dengan pada saat mengenkrpsi data. Klik tombol "Dekripsi" dan file akan diproses.



Gambar 4.8. Tampilan *Browsing File*



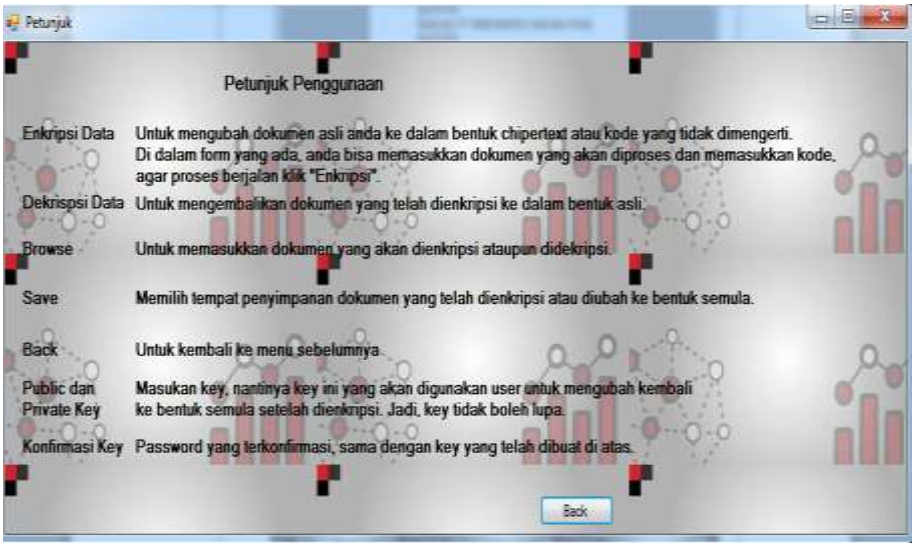
Gambar 4.9. Tampilan proses dekripsi



Gambar 5.0. Tampilan *file* yang telah didekripsi

4. Tampilan Petunjuk Penggunaan

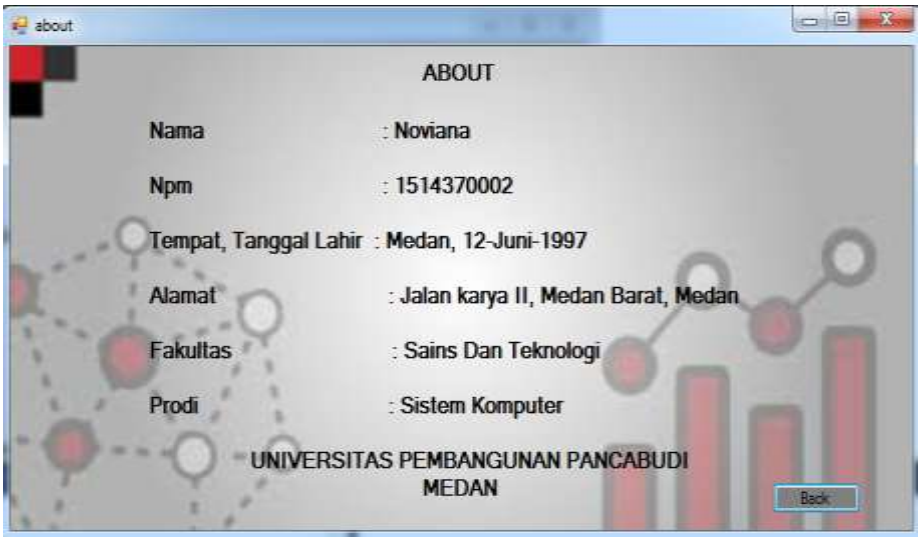
Tampilan ini berisi tentang petunjuk penggunaan aplikasi dan keterangan untuk setiap perintah yang ada.



Gambar 5.1. Tampilan Petunjuk Penggunaan

5. Tampilan *About*

Tampilan yang berisi tentang biodata *creator*.



Gambar 5.2. Tampilan About

BAB V
Penutup

5.1 Simpulan

Berdasarkan hasil studi literatur, analisis, perancangan, implementasi, dan pengujian sistem yang telah dilakukan. Maka, kesimpulan yang didapat adalah sebagai berikut:

1. Aplikasi ini mampu untuk mengamankan file dokumen dan cukup bagus digunakan untuk mahasiswa / mahasiswi yang membutuhkan pengamanan dokumen yang sederhana namun terjamin keamanannya.
2. Hasil enkripsi berupa kode ASCII yang sulit diterjemahkan.
3. Kunci yang dimasukkan oleh pengguna secara otomatis mampu dibaca oleh sistem, sehingga mudah bagi pengguna dalam memproses enkripsi dan dekripsi data.

5.2 Saran

Saran - saran untuk penelitian dan perkembangan penelitian selanjutnya adalah sebagai berikut:

1. Aplikasi yang ada tidak hanya berbasis *desktop* tapi juga mampu dikembangkan aplikasi yang berbasis *mobile*.
2. Sistem ini hanya menggunakan satu kunci keamanan, untuk penelitian selanjutnya dapat dikembangkan untuk mengkombinasi dua kunci agar lebih terjaga keamanannya.

DAFTAR PUSTAKA

Arif, A., Saputra, R, 2016. “Implementasi Kriptografi Kunci Publik Dengan Algoritma RSA-CRT Pada Aplikasi Instant Messaging”, Jurnal Ilmiah Informatika, Volume 3, Nomor.1

Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Fachri, Barany. "Aplikasi Perbaikan Citra Efek Noise Salt & Papper Menggunakan Metode Contraharmonic Mean Filter." Seminar Nasional Royal (Senar). Vol. 1. No. 1. 2018.

File”, Seminar Nasional Royal, Hal 1-6

Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. Int. J. Recent Trends Eng. Res, 3(8), 58-64.

Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). Implementasi Augmented Reality Sebagai Media Promosi Penjualan Rumah. In Seminar Nasional Royal (Senar) (Vol. 1, No. 1, pp. 429-434).

Kristanto, Rivalri Hondro., dan Widi, Gunadi Nurcahyo, 2014. “Analisis Perancangan Sistem Yang Menerapkan Algoritma Triangle Chain Chiper(TCC) Untuk Enkripsi Record Tabel Database”, Jurnal Teknologi Informasi dan Komputer, Volume 3, Nomor.2 Hal 118-127

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." INTECOMS: Journal of Information Technology and Computer Science 1.1 (2018): 72-77.

Rachman, Arif Kurnia, 2010. "Perbandingan Mode Chiper Electronic Code Book dan Chiper Block Chaining Dalam Pengamanan Data", Jurnal Teknologi, Volume 3, Nomor.1 Hal 84-89

Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., ... & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In Journal of Physics: Conference Series (Vol. 1028, No. 1, p. 012052). IOP Publishing.

Sadikin, Rifki, 2018. "Kriptografi Untuk Keamanan Jaringan" Yogyakarta, Andi

Safrina, Nanda, 2017. "Implemetasi Kriptografi Hybrid Algoritma Elgamal dan Double Playfair Chiper Dalam Pengamanan File Jpeg Berbasis Desktop"

Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Int. J. Sci. Res. Sci. Technol, 3(6), 470-473.

Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.

Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." Int. J. Eng. Trends Technol 42.5 (2016): 218-222.

Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

Suyuti, Muhammad Ma'sum., Azhar M Irwansyah, Priyanto, H, 2017. "Analisis Perbandingan Sistem Keamanan Snort dan Netfilter", Jurnal Sistem dan Teknologi Informasi, Volume 5, Nomor. 1 Hal 56-60

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.

Wahyuni, Sri., Lubis, Akhyar., Batubara, Supina., Kamil, Iqbal Siregar, 2018.