IMPLEMENTASI ALGORITMA BLOWFISH DAN RSA

UNTUK ENKRIPSI DAN

DEKRIPSI FILE

SKRIPSI

DiajukanSebagai Salah SatuSyaratUntukMemperolehGelar

SARJANA TEKNIK

Program StudiTeknikInformatika – Strata 1

OLEH:

NAMA : RIKI ISMAYANA

NIM : 011201573125003

FAKULTAS : TEKNIK

UNIVERSITAS SATYA NEGARA INDONESIA

JAKARTA

2016

SURAT PERNYATAAN KARYA SENDIRI

Yang bertandatangan dibawah ini :

Nama	:	RIKI ISMAYANA
NIM	:	011201573125003
Program Studi	:	TEKNIK INFORMATIKA

Menyatakan bahwa Skripsi ini adalah murni hasil karya sendiri dan seluruh isi Skripsi menjadi tanggung jawab saya sendiri. Apabila saya mengutip dari karya orang lain maka saya mencantumkan sumbernya sesuai dengan ketentuan yang berlaku. Saya bersedia dikenai sanksi pembatalan Skripsi ini apabila terbukti melakukan tindakan plagiat (penjiplakan)

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 17 Februari 2016

(RikiIsmayana)

NIM: 011201573125003

LEMBAR PENGESAHAN SKRIPSI

NAMA	: Riki Ismayana
NIM	: 011201573125003
JURUSAN	: Teknik Informatika
KOSENTRASI	: Rekayasa Perangkat Lunak
JUDUL SKRIPSI	: Implementasi Algoritma Blowfish dan RSA Untuk Enkripsi dan Dekripsi File
TANGGAL UJIAN	: 17 Februari 2016

JAKARTA, 17 Februari 2016

Dosen Pembimbing II	Dosen Pembimbing I
Faizal Zuli, S.Kom., M.Kom., M.T.A	**Berlin Sitorus, S.Kom.,M.Kom.**

Dekan	Ketua Program Studi
Ir. Nurhayati, M.Si	**Safrizal, ST., M.M., M.Kom**

LEMBAR PENGESAHAN PENGUJI

IMPLEMENTASI ALGORITMA BLOWFISH DAN RSA UNTUK ENKRIPSI DAN DEKRIPSI FILE

OLEH :

NAMA : RIKI ISMAYANA

NIM : 011201573125003

Telah dipertahakan didepan Penguji pada tanggal 17 Februari 2016

Dan dinyatakan telah memenuhi syarat untuk diterima

Ketua Penguji

Bosar Panjaitan, S.SI., M.Kom

Anggota Penguji I **Anggota Penguji II**

Berlin Sitorus, S.Kom.,M.Kom **Agung Priambodo, S.Kom.,M.Kom**

KATA PENGANTAR

Bismillahirrahmanirrahim

Puji dan syukur saya panjatkan kepada Allah SWT, karena dengan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul **"IMPLEMENTASI ALGORITMA BLOWFISH DAN RSA UNTUK ENKRIPSI DAN DEKRIPSI FILE"**

Dalam penulisan skripsi ini tidak lepas bantuan dari berbagai pihak, baik secara moril maupun materil, untuk itu penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

(a) Ibu Ir. Nurhayati, M.Si selaku Dekan Fakultas Teknik Universitas Satya Negara Indonesia.

(b) Bapak Safrizal, ST., MM.,M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Satya Negara Indonesia.

(c) Bapak Berlin Sitorus, S.Kom.,M.Kom. selaku Pembimbing I dan Bapak FaizalZuli,S.Kom, M.KOM, MTA selaku Pembimbing II yang memberikan masukan serta inputan pada saat bimbinganskripsi.

(d) Bapak. HernalomSitorus, S.Kom, M.Kom yang banyakberjasakepadapenulisselamakuliah di USNI.

(e) Bapak Bosar Panjaitan, S.SI.,M.Kom selaku ketua penguji serta Bapak Agung Priambodo, S.Kom.,M.Kom dan Bapak Berlin Sitorus, S.Kom.,M.Kom. selaku anggota penguji yang telah memberikan banyak masukan pada saat pengujian skripsi.

(f) Kedua orang tua (Ibu Euis Suhartini dan Bapak Sunarmo) yang selalu mendoakan tiada henti dan mensupport penulis.

(g) Istritercinta, WasiaturRizqi.

(h) Adik-adikkutersayang, JokoWijanarko, PutriNurhidayah, danFitrahNurdiansah.

(i) Seluruhteman-temanseperjuangan (GalihMahendra, Pak Dhe, April Wibowo, Evan Abieza, BrajaSantosadanlain-lain)

(j) Serta semua pihak yang terlibat secara langsung maupun tidak langsung yang penulis tidak bisa sebutkan satu persatu.

Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi para pembaca dan pihak yang berkepentingan.

Jakarta, 17 Februari 2016

Penulis

ABSTRACK

Implementation Blowfish algorithm and RSA for Encryption and Decryption File

Under the guidance of **Berlin Siotrus, S. Kom, M. Kom** and **Faizal Zuli, Kom., M. Kom, MTA.**

This research is motivated by the problems in the .txt file is a text file format that is very popular in the world of computers. This format can be run on various operating systems such as Windows, Linux, MacOS, and others. In the windows operating system .txt format can be opened with Notepad and WordPad program. .txt File one of which is used in the game as a set of commands to be executed when playing games, such as: raising the level, raises the boss, get an item in the game, and much more. To protect .txt files contained on a computer or data storage from unauthorized access one way is by using cryptography. Blowfish is one of the algorithms are not patentable and strong enough because it has a large key space and the length can vary, so it is not vulnerable on the key. A cryptographic system that is well located on the secrecy of the key and not on secrecy algorithm used. RSA (Rivest Shamir Adleman) is an abbreviation of the authors is that Ron Rivest, Adi Shamir, and Leonard Adleman created in 1977. RSA is an asymmetric cryptography system. The purpose of this study was able to design and implement the Blowfish algorithm and RSA cryptography application file. The benefits of the design and implementation of these applications are able to secure and maintain the confidentiality of data using Blowfish cryptography and RSA as well as a reference for research related to cryptography next.

Keywords: cryptography, Blowfish, RSA

ABSTRAK

Implementasi AlgoritmaBlowfish dan RSA untuk Enkripsi dan DekripsiFile

Dibawah bimbingan **Berlin Siotrus, S.Kom, M.Kom** dan **Faizal Zuli, S.Kom., M.Kom, MTA.**

Penelitian ini dilatarbelakangi oleh permasalahan file .txt merupakan format file teks yang sangat populer dalam dunia komputer. Format ini bisa dijalankan di berbagai sistem operasi seperti windows, linux, macos, dan lain-lain. Dalam sistem operasi windows format .txt bisa dibuka dengan program notepad dan wordpad. File .txt salah satunya digunakan dalam game sebagai kumpulan perintah yang akan dijalankan ketika bermain game, diantaranya: menaikkan level, memunculkan boss monster, mendapatkan item di dalam game, dan masih banyak lagi. Untuk melindungi file .txt yang terdapat pada komputer atau tempat penyimpanan data dari akses illegal salah satu caranya adalah dengan menggunakan kriptografi. 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. Tujuan dari penelitian ini adalah dapat merancang dan mengimplementasikan algoritmaBlowfish dan RSA pada aplikasi kriptografi file. Adapun manfaat dari perancangan dan implementasi aplikasi ini adalah dapat mengamankan dan menjaga kerahasiaan data dengan menggunakan kriptografi Blowfish dan RSA serta sebagai referensi untuk penelitian yang berhubungan dengan kriptografi selanjutnya.

Kata Kunci: Kriptografi, Blowfish, RSA

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

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.

Dalam kriptografi terdapat beberapa metode yang cukup penting dalam pengamanan data, untuk menjaga kerahasiaan data salah satunya adalah enkripsi (encryption). Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi chipertext. Sedangkan suatu proses yang dilakukan untuk mengubah pesan tersembunyi menjadi pesan asli disebut dekripsi. Pesan biasa atau pesan asli disebut plaintex sedangkan pesan yang telah diubah atau disandikan supaya tidak mudah dibaca disebut dengan chipertext.

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.

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.

Blowfish pada strategi implementasi yang tepat akan lebih optimal, dapat berjalan pada memori kurang dari 5 KB dan kesederhanaan pada algoritmanya. Untuk itu dibangun sebuah aplikasi yang dapat digunakan untuk mengamankan data atau informasi berupa file dengan menggunakan metode Blowfish ini. Selain itu diharapkan pula aplikasi yang dibangun ini dapat melihat kinerja algoritma blowfish dari segi waktu prosesnya.

1.2 Rumusan Masalah

Rumusan masalah dalam pembuatan dan implementasi aplikasi ini adalah bagaimana mengimplementasi algoritma blowfish dengan kunci asimetris untuk enkripsi dan dekripsi file**.**

1.3 Ruang Lingkup Penulisan

Dalam pembangunan dan implementasi aplikasi ini, penulis membatasi permasalahan tentang:

a. Merancang aplikasi enkripsi dan dekripsi menggunakan teknik kunci asimetris algoritma blowfish dengan objek yang berupa file text.

b. Ukuran maksimal file text yang digunakan adalah 5 *Megabyte*.

c. Merancang enkripsi dan dekripsi chipertext dengan algoritma RSA.

1.4 Tujuan dan Manfaat Penulisan

1.4.1. Tujuan

Adapun tujuan yang ingin dicapai penulis ini adalah dapat merancang dan mengimplementasikan teknik kunci asimetris algoritma blowfish pada aplikasi kriptografi file.

1.4.2. Manfaat

Adapun manfaat dari perancangan dan implementasi aplikasi ini adalah sebagai berikut :

a) Mengamankan dan menjaga kerahasiaan data dengan menggunakan aplikasi kriptografi blowfish.

b) Dapat menyamarkan file input dan mengembalikannya kembali menjadi file semula (enkripsi dan dekripsi dapat bekerja)

c) Sebagai referensi untuk penelitian yang berhubungan dengan algoritma blowfish lebih lanjut.

1.5 Sistematika Penulisan

Adapun susunan penulisan laporan ini, penyusunannya diuraikan dalam beberapa bab yaitu sebagai berikut :

BAB I **: PENDAHULUAN**

Berisikan tentang latar belakang, rumusan masalah, ruang lingkup penulisan, tujuan dan manfaat penulisan, dan sistematika penulisan.

BAB II **: LANDASAN TEORI**

Bab ini berisikan hasil penelitian – penelitian yang telah dilakukan dan dasar – dasar teori guna untuk sebagai pedoman, acuan dan penunjang dalam penyelesaian masalah.

BAB III **: METODE PENELITIAN**

Bab ini berisi mengenai metode penelitian yang digunakan dengan menggunakan metode waterfall.

BAB IV **: PERANCANGAN SISTEM**

Bab ini berisi tentang penjelasan mengenai analisis dan desain perancangan dari aplikasi yang dibuat.

BAB V **: HASIL DAN PEMBAHASAN**

Bab ini berisi tentang pengujian sistem, perangkat lunak, dan perangkat keras yang digunakan.

BAB VI **: KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari aplikasi yang dibuat dan saran – saran yang bermanfaat dalam pengembangan aplikasi lebih lanjut.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Menurut Bruce Schneier, kriptografiadalah ilmu dan seni untuk menjaga keamanan pesan. Kata “seni” dalam definisitersebut berasal dari fakta sejarah bahwa pada masa-masa awal sejarahkriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakanpesan. Cara-cara unik tersebut mungkin berbeda-beda pada setiap pelakukriptografi sehingga setiap cara menulis pesan rahasia, pesan mempunyai nilaiestetika tersendiri sehingga kriptografi berkembang menjadi sebuah senimerahasiakan pesan (kata “graphy” di dalam “cryptography” itu sendiri sudahmenyiratkan sebuah seni).

2.2 Landasan teori

2.2.1 Aplikasi

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

2.2.2 Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu cryptós yang artinya “secret” (yang tersembunyi) dan grápheinyang artinya “writting” (tulisan). Jadi, kriptografi berarti ”secret writting” (tulisan rahasia). Definisi yangdikemukakan 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) menyampaikantujuan kriptografi bahwa untuk memberi layanan keamanan (yang juga dinamakansebagai aspek-aspek keamanan) sebagai berikut :

a. Kerahasiaan (confidentiality), adalah layanan yang ditujukan untukmenjaga agar pesan tidak dapt dibaca oleh pihak-pihak yang tidak berhak.Di dalam kriptografi, layanan ini direalisasikan dengan menyandikanpesan menjadi chiperteks.
b. Integritas data (data integrity), adalah layanan yang menjamin bahwapesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.Dengan kata lain, aspek keamanan ini dapat diungkapkan sebagaipernyataan: “Apakah pesan yang diterima masih asli atau tidak mengalamiperubahan (modifikasi)?”. Untuk menjaga integritas data, system harusmemiliki kemampuan untuk mendeteksi manipulasi pesan oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, danpensubstitusian data lain ke dalam pesan yang sebenarnya.

Kriptografi sendiri mempunyai komponen-komponen untuk mencapai tujuan kriptografi. Menurut Ariyus (2006: 19), pada dasarnya kriptografi terdiri daribeberapa komponen seperti :

a. Enkripsi : Enkripsi merupakan hal yang sangat penting dalam kriptografisebagai pengamanan atas data yang dikirimkan agar rahasianya terjaga.Pesanaslinya disebut plainteks yang diubah menjadi kode-kode yang tidakdimengerti.Enkripsi bisa diartikan sebagai

chiperatau kode. Seperti ketikakita tidakmengerti akan artisebuah kata, kita bisa melihatnya di dalamkamus atau daftaristilah. Berbeda dengan enkripsi, untuk mengubahplainteks ke bentukchiperteks digunakan algoritma yang bisa mengkodekandata yang diinginkan.

b. Dekripsi : Dekripsi merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (Plainteks), yang disebut dekripsipesan. Algoritma yang digunakanuntuk dekripsi tentu berbeda dengan yangdigunakan untuk enkripsi.

c. Kunci : Kunci yang yang dimaksud di sini adalah kunci yang dipakai untukmelakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, yakni kuncipribadi (private key) dan kunci umum (public key).

d. Chiperteks : merupakan suatu pesan yang sudah melalui proses enkripsi. Pesanyang ada pada chiperteks tidak bisa dibaca karena berisi karakter-karakter yangtidak memiliki makna (arti).

e. Plainteks : sering juga disebut cleartext; merupakan suatu pesan bermakna yang ditulis atau diketik dan plainteks itulah yang akan diproses menggunakanalgoritma kriptografi agar menjadi chiperteks.

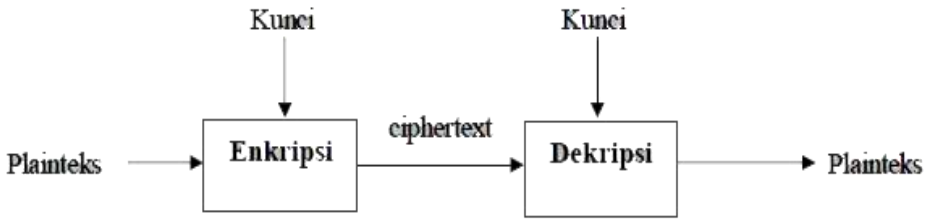
f. Pesan : pesan bisa berupa data atau infomasi yang dikirim (melalui kurir, salurankomunikasi data, dan sebagainya) atau yang disimpan di dalam mediaperekaman (kertas, storage, dan sebagainya).

g. Cryptanalysis : bisa diartikan sebagai analisis sandi atau suatu ilmu untukmendapatkan plainteks tanpa harus mengetahui kunci secara wajar. Jika suatuchiperteks berhasil menjadi plainteks tanpa menggunakan kunci yang sah, makaproses tersebut dinamakan breaking code yang

dilakukan oleh para cryptaanalys.Analisis sandi juga mampu menemukan kelemahan dari suatu algoritmakriptografi dan akhirnya bisa menemukan kunci atau plainteks dari chiperteksyang dienkripsi menggunakan algoritma tertentu.

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 kerahasiaannyaterbongkar maka isi pesan akan terbongkar.

Berikut adalah skema yangmengilustrasikan enkripsi dan dekripsi.



Gambar 2.1Skema Enkripsi dan Dekripsi

(Munir, 2006: 6)

Pada gambar 2.1 mengilustrasikan sebuah plainteks dienkripsi menggunakankunci enkripsi sehingga menjadi chiperteks dan

chiperteksdidekripsi kembalimenggunakan kunci dekripsi sehingga menjadi plainteks kembali.

Secara formal proses enkripsi dapat dirumuskan sebagai berikut :

$$\mathbf{E_k(P) = C} \qquad (2.1)$$

Dimana E merupakan fungsi enkripsi dengan menggunakan k (kunci) beroperasipada P (plainteks) sehingga menghasilkan C (chiperteks).Sedangkan pada prosesdekripsi dirumuskan dengan :

$$\mathbf{D_k(C) = P} \qquad (2.2)$$

Dimana D merupakan fungsi enkripsi dengan menggunakan k (kunci) beroperasipada C (chiperteks) sehingga menghasilkan P (plainteks).
Sehingga kedua hubungantersebut menjadi :

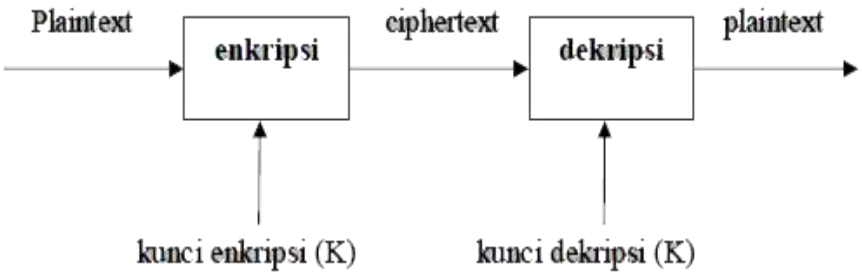
$$\mathbf{D_k(Ek(P)) = P} \qquad (2.3)$$

2.2.3 Algoritma Berdasarkan Jenis Kunci

Algoritma kriptografi berdasarkan jenis kunci terbagi menjadi dua kategoriyaitu algoritma simetris dan algoritma asimetris. Berikut adalah penjelasanalgoritma simetis dan asimetris.

a. Algoritma Simetris

Algoritma simetris adalah algoritma kriptografi dimana kunci enkripsidandekripsi sama. Contoh dari algoritma simetris adalah blowfish,rijndael, camellia, danlain-lain. Berikut skema dari algoritma kuncisimetris.



Gambar 2.2 Diagram Proses Enkripsi dan Dekripsi Algoritma Simetris

Pada gambar 2.2 merupakan diagram proses enkripsi dan dekripsi padaalgoritma simetris. Pada proses tersebut plainteks dienkripsi menggunakan kuncienkripsi (K) sehingga menghasilkan chiperteks. Chiperteks didekripsikan kembalimenggunakan kunci dekripsi (K), artinya kunci pada dekripsi sama dengan kunciketika pengenkripsian dilakukan sehingga menghasilkan plainteks.

Kelebihan :

a. Kecepatan operasi lebih tinggi bila dibandingkan dengan algoritmaasimetrik.

b. Karena kecepatannya yang cukup tinggi, maka dapat digunakan pada sistem real-time.

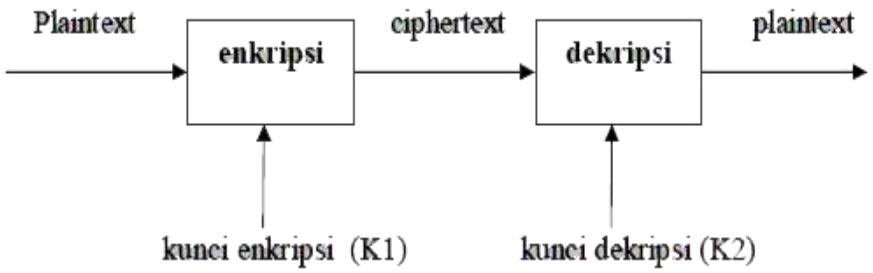
Kelemahan :

a. Untuk tiap pengiriman pesan dengan pengguna yang berbeda dibutuhkankunci yang berbeda juga, sehingga akan terjadi kesulitan dalammanajemen kunci tersebut.

b. Permasalahan dalam pengiriman kunci itu sendiri yang disebut "keydistribution problem"

b. Algoritma Asimetris

Algoritma asimetris adalah suatu algoritma kriptografi dimana kunciuntuk enkripsi yang digunakan berbeda dengan kunci dekripsi. Kuncienkripsidinamakan sebagai kunci public yaitu kunci yang bebasdiketahui oleh siapapun,sedangkan kunci dekripsi dinamakan kuncikunci privat yaitu kunci yang hanyaboleh diketahui oleh penerimapesan. Contoh dari algoritma kriptografi asimetris adalah RSA,lgamal,dan lain-lain. Berikut adalah skema dari algoritmaasimetris:



Gambar 2.3 Diagram Proses Enkripsi dan Dekripsi Algoritma Asimetris

Pada gambar 2.3 merupakan diagram proses enkripsi dan dekripsi padaalgoritma asimetris. Pada proses tersebut plainteks dienkripsi

menggunakan kuncienkripsi (K1) sehingga menghasilkan chiperteks. Chiperteks didekripsikankembali menggunakan kunci dekripsi (K2), artinya kunci pada saat pendekripsianberbeda dengan kunci ketika pengenkripsian dilakukan sehingga menghasilkanplainteks.

Kelebihan :

a. Masalah keamanan pada distribusi kunci dapat lebih baik

b. Masalah manajemen kunci yang lebih baik karena jumlah kunci yang lebihsedikit

Kelemahan :

a. Kecepatan yang lebih rendah bila dibandingkan dengan algoritma simetris

b. Untuk tingkat keamanan sama, kunci yang digunakan lebihpanjang dibandingkan dengan algoritma simetris.

2.2.3 Metode Algoritma RSA

Membahas prinsip kerja RSA, serta berbagai teori dan perhitungan matematika yang digunakan pada algoritma RSA.

a. Prinsip Kerja RSA

RSA (Rivest Shamir Adleman) adalah singkatan dari para pembuatnya yaitu Ron Rivest, Adi Shamir dan Leonard Adleman yang dibuat pada tahun 1977. RSA merupakan sistem kriptografi asimetrik.

Prinsip kerja RSA hanya menggunakan operasi pemangkatan dan operasi mod (modulus) yang menghasilkan nilai yang relatif acak.

$C = M^e \bmod n$ (2.1)

Pada tabel 2.1 terlihat tingkat kesulitan untuk menemukan hubungan yang simetris antara angka-angka dikolom M dengan angka-angka dikolom C, dikolom C diperoleh dari operasi M^7 mod 77.

Tabel 2.1 Nilai $C = M^7$ mod 77

M	2	3	4	5	6	7	8	9	10
C	51	31	60	47	41	28	57	37	10

Karena hubungan yang terlihat acak, maka sulit untuk mendapatkan kembali nilai M walaupun diketahui nilai e, n dari hasil operasi M^e mod n. Oleh karena itu dibutuhkan nilai pasangan dari e pada contoh tersebut pasangan dari e = 7 dan e = 43, tepatnya dilakukan M^7 mod 77 lalu hasilnya dipasangkan dengan 43 dan di-mod-kan dengan 77, maka akan didapat nilai M kembali secara matematis dapat ditulis :

$M = C^d \bmod n$ (2.2)

Contoh:

jika C = 31, d = 43 dan n = 77 maka:

$M = 31^{43} \bmod 77$

$= 3$

Pada prinsipnya kerja RSA ini adalah dimana e sebagai kunci publik, d sebagai kunci pribadi, M sebagai plaintext, C sebagai

ciphertext, dan n sebagai modulus publik. Dari prinsip kerja RSA dapat dilihat bahwa keamanan sistem penyandian RSA bergantung pada kunci-kunci yang digunakan untuk enkrip dan deskripsi, ukuran kunci yang digunakan pada sistem penyandian ini menentukan jumlah kombinasi kunci yang mungkin. Jika menggunakan ukuran 64 bit, maka total banyaknya kombinasi kunci yang mungkin adalah 2^{64} = 18446744073709551616

Ukuran yang terbaik untuk modulus publik RSA bergantung pada kebutuhan keamanan, jika modulusnya besar, maka akan diperoleh keamanan yang lebih terjamin, tetapi kecepatan operasi RSA nya lebih lambat, karena untuk membangkitkan modulus n harus merupakan hasil kali dua bilangan prima p dan q yang menyusun modulus harus sama panjangnya. Hal ini akan membuat modulus lebih kokoh. Jadi jika digunakan modulus 64 bit maka masing-masing bilangan prima harus mempunyai panjang 32 bit, dua bilangan prima yang menyususn modulus n berasal dari bilangan-bilangan acak yang terdapat pada program, dengan metode pembangkitan bilangan acak untuk menentukan bilangan prima p dan q.

b. Teori Matematika

Teori dasar secara matematika yang diterapkan dalam pengamanan data digital dengan teknik enkripsi RSA.

1. Bilangan Prima

Merupakan bilangan bulat utuh (integer) yang lebih besar dari satu serta hanya dapat dibagi oleh faktor satu dan bilangan itu sendiri.

Jika dua bilangan adalah prima secara relatif pada saat kedua bilangan tersebut tidak membagi faktor yang bersamaan selain dari bilangan satu. Dengan kata lain misalnya p adalah bilangan prima relatif terhadap n jika pembagi bersama terbesar (greatest common divisor) dari p dan n sama dengan satu, secara matematis ditulis sebagai.

$$GCD(p,n)=1 \qquad (2.3)$$

2. Algoritma Euclidean

Algoritma Euclidean (*Euclid's algorithm*) adalah suatu proses untuk menemukan pembagi bersama terbesar (greatest common divisor) dari dua bilangan bulat untuk semua pasangan dari bilangan bulat utuh (integer) yaitu s dan d terdapat sepasang bilangan bulat utuh yang unik, yaitu q sebagai hasil bagi dan r sebagai sisa, secara matematis dapat ditulis sebagai berikut.

$$s = dq + r \qquad (2.4)$$

Dengan $0 \leq r \leq |d|$

Jika diberikan dua bilangan bulat positif r dan s pembagi bersama terbesarnya dapat dihitung dengan penerapan algoritma pembagian euclidean pada saat r < s, algoritma euclidean menggunakan langkah-langkah berikut.

$s = q_1 r + r_1$ (2.5)

$r = q_2 r + r_2$ (2.6)

$r_1 = q_3 r + r_3$ (2.7)

.

.

.

$r_{n-2} = q_n r_{n-1} + r_n$ (2.8)

$r_{n-1} = q_{n+1} r_n$ (2.9)

proses ini akan berhenti pada saat telah didapatkan sisa sama dengan 0. sisa rn yang bukan nol terakhir adalah pembagi bersama terbesar (greatest common divisor) dari r dan s. secara matematika dapat ditulis sebagai berikut.

$Rn = GCD(r,s)$ (2.10)

Dari algoritma pembagian euclidean dapat diperoleh beberapa sifat aljabar bilangan bulat utuh yang sangat penting, yaitu untuk setiap bilangan bulat utuh r dan s terdapat bilangan bulat utuh a dan b yang disebut kombinasi linear dari r dan s secara matematis dapat ditulis sebagai berikut.

$GCD(r,s) = ar + bs$ (2.11)

3. Teorema Euler dan Fungsi -ϕ Euler

leonhard pada tahun 1760 membuat teorema euler dan fungsi -ϕ euler. jika GCD(a,n) = 1, maka aϕ(n) = 1 (mod n). Teorema euler ini berguna dalam perumusan pangkat yang besar dari modulus n.

3.1 Fungsi -ϕ Euler

Fungsi -ϕ atau euler's totient function di definisikan sebagai ϕ(n) adalah jumlah bilangan bulat utuh (integer) positif yang tidak lebih dari n dan merupakan bilangan prima secara relatif dengan n utuk n ≥ 1

jika n bilangan prima, maka setiap bilangan bulat utuh positif yang kurang dari n adalah prima secara relatif dengan n. Secara matematik dapat ditulis.

$$\phi(n) = n - 1 \qquad (2.12)$$

jika dan hanya jika n adalah bilangan prima

contoh : ϕ(11) = 10, ϕ(23) = 22, ϕ(29) = 28

jika n = p* q dengan p dan q adalah bilangan prima.

$$\phi(n) = \phi(p) * \phi(q) \qquad (2.13)$$

$$\phi(n) = (p-1) * (q-1) \qquad (2.14)$$

jika dan hanya jika p dan q adalah bilangan prima

contoh : ϕ(35)

$$\phi(35) = \phi(5) * \phi(7) = (5-1) * (7-1)$$

$$= 4 * 6 = 24$$

Jadi bilangan bulat utuh positif 24 adalah prima secara relatif dengan 35

3.2 Teorema Euler

Misalnya $r_1,r_2,...,r_{\phi(n)}$ menjadi bilangan bulat utuh positif $\phi(n)$ yang kurang dari n dan merupakan bilangan prima secara relatif dengan n. Jika diberikan suatu nilai a, maka $ar_1,ar_2,...,ar_{\phi(n)}$ harus kongruen dengan perubahan urtutan angka dari $r_1,r_2,...,r_{\phi(n)}$ (mod n) pada urutan tertentu, dan bilangan-bilangan $\phi(n)$ semuanya harus berbeda dan prima secara relatif dengan n.

Untuk ar_i, $r_i \varepsilon z$ dapat ditentukan bahwa ar_i adalah kongruen (sama dan sebangun) dengan r_i (mod n) yang dilambangkan dengan.

$$ar_i \equiv r_i \pmod{n} \qquad (2.15)$$

jika ari – ri dapat dibagi rata dengan n, sehingga

$$ar_i - r_i = n * t \qquad (2.16)$$

untuk $t \varepsilon z$

contoh : $17 \equiv 33 \pmod 8$, karena $17 - 33 = 8(-2)$

Hal ini berarti bahwa ari dan ri mempunyai sisa yang sama pada saat dibagi dengan n tetapi ari tidak perlu lebih kecil dari n.

Pada saat $GCD(ar_i,n) = 1$ untuk setiap i, $1 \leq i \leq \phi(n)$ maka berlaku persamaan berikut [7]

$$(ar_1)\,(ar_2)\,...\,(ar_{\phi(n)}) \equiv r_1\ r_2\ ...\ r_{\phi(n)} \pmod{n} \qquad (2.17)$$

atau

$$a^{\phi(n)}\,(r_1\,r_2\,...\,r_{\phi(n)}) \equiv r_1\,r_2\,...\,r_{\phi(n)} \pmod{n} \qquad (2.18)$$

pada saat GCD(r_i,n) = 1 dengan i = 1, 2, ...,ϕ(n), hal ini menunjukkan bahwa hasil kali r_1, r_2, ...,$r_{\phi(n)}$ juga adalah bilangan prima secara relatif dengan n, sehingga GCD(r_1 r_2 ... $r_{\phi(n)}$, n) = 1

jika n $\geq$ 2 adalah bilangan bulat utuh (integer) positif dan GCD(a,n) $\equiv$ 1 maka.

$a^{\phi(n)} = 1 \pmod{n}$ (2.19)

dengan ϕ(n) merupakan bilangan bulat utuh positif kurang dari n dan prima secara relatif dengan n. Persamaan (2.19) disebut teorema euler dengan fungsi -ϕ euler.

4. Algoritma RSA

Model sistem penyandian RSA memiliki 2 buah kunci yang berbeda, yaitu enkripsi dan kunci deskripsi. Enkripsi untuk menyandikan sebuah pesan (plaintext) adalah bilangan bulat utuh (integer) disimbolkan e dan kunci diskripsi menerjemahkan pesan tersandi (ciphertext) adalah bilangan bulat utuh yang disimbolkan dengan d .

Dalam perancangan ini terdapat bilangan bulat untuh n sebagai modulus publik yang diperoleh dari perkalian dua bilangan prima secara acak yaitu p dan q dan p $\neq$ q.

n = p * q (2.20)

kunci e sebagai kunci publik (publik key) dipilih sebagai bilangan prima dan prima secara relatif (reletively prime) dengan ϕ(n) sebagai euler's untuk totient function.

$\phi(n) = (p - 1) * (q - 1)$ (2.21)

Kunci e adalah prima secara relatif dengan $\phi(n)$ jika

$$GCD\ (e, \phi(n)) = 1 \qquad (2.22)$$

Yang dapat dibuktikan dengan algoritma Euclidean.

Ciphertext C dapat dihasilkan dengan cara memproses plaintext M, $0 \leq M \leq n-1$ yang dapat dipangkatkan dengan kunci publik e kemudian dimod-kan dengan modulus publik n, seperti yang diperlihatkan berikut ini.

$$E_k\ (M) = C = M^e\ (\text{mod } n) \qquad (2.23)$$

Sebaliknya plaintext asli dapat dihasilkan kembali dengan memproses ciphertext C yang dipangkatkan dengan kunci pribadi d kemudian di-mod-kan dengan modulus publik n, seperti.

$$D_k\ (C) = M = C^d\ (\text{mod } n) = M^{ed}\ (\text{mod } n) \qquad (2.24)$$

Kunci pribadi (private key) d dapat dihasilkan dengan menggunakan teorema euler yang dapat menurunkan pangkat yang besar dari modulus n dengan urutan sebagai berikut. Dari teorema euler $a^{\phi(n)} \equiv 1\ (\text{mod } n)$ untuk GCD(a,n) = 1, misalkan a menjadi plaintext maka teorema euler menjadi.

$$M^{\phi(n)} \equiv 1\ (\text{mod } n) \qquad (2.25)$$

Dengan GCD(m,n) = 1 sehingga plaintext M, adalah prima secara relatif dengan n, berdasarkan teorema

$$a^{\lambda\phi(n)} \equiv 1^{\lambda}\ (\text{mod } n) \qquad (2.26)$$

Jika $a = b\ (\text{mod } n)$ maka berlaku $a\mu = b\mu\ (\text{mod } n)$ untuk $GCD(\mu,n) = 1$. dari teorema tersebut maka persamaan (2.26) menjadi.

$M^{\lambda\phi(n)} * M \equiv 1 * M \pmod{n}$.

$M^{\lambda\phi(n)+1} \equiv 1 * M \pmod{n}$ (2.27)

Dari persamaan (2.24) dan (2.27) dapat dilihat

$\lambda\phi(n)+1 = e * d$

$1 = e * d - \lambda\phi(n)$ (2.28)

Persamaan (2.28) dapat ditulis dalam bentuk

$e * d \equiv 1 \pmod{\phi(n)}$ (2.29)

yang merupakan fungsi matematik untuk menghitung kunci pribadi d berdasarkan teorema euler serta dipengaruhi oleh kombinasi linear dari e dan $\phi(n)$ menghasilkan bentuk $1 = e * d - \lambda\phi(n)$.

Jika kombinasi linear dari e dan $\phi(n)$ menghasilkan bentuk

$1 = \lambda\phi(n) – e * d$ (2.30)

maka untuk memperoleh bentuk $1 \equiv e * d \pmod{\phi(n)}$ nilai d harus ditempatkan kembali dengan $\phi(n) – d$, sehingga persamaan (2.30) menjadi.

$1 = \lambda\phi(n) – e(\phi(n) – d)$

$1 = \lambda\phi(n) – e\phi(n) + ed$

$1 = (\lambda – e)\ \phi(n) + ed$ (2.31)

dengan λ - e adalah bilangan bulat utuh, dengan kata lain, jika nilai $d < 0$ maka nilai d menjadi $d1 = d + \phi(n)$.

2.2.4 Algoritma Berdasarkan Mode Bit

Berdasarkan mode bit yang diolah dalam satu kali proses, maka algoritmakriptografi dapat dibedakan menjadi dua jenis yaitu algoritma blok chiper danstream chiper. Algoritma blok chiper dibagi menjadi beberapa mode blok yaitumode Electronic Code Book (ECB), Chiper Blok Chaining (CBC), Chiper Feed Back (CFB), dan Output Feed Back (OFB). Sedangkan algoritma stream chiperdibagi menjadi dua yaitu Synchronous Stream Chiper dan Self-Synchronous Stream Chiper.

a. Algoritma Blok Cipher

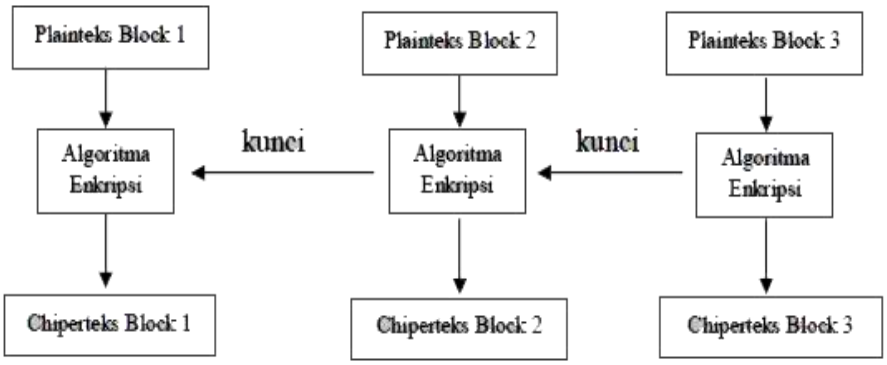
″ Blok chipher merupakan suatu algoritma yang mana input dan outputnyaberupa satu blok, dan setiap blok terdiri dari beberapa bit (1 blok terdiri dari 64bit atau 128 bit)". (Ariyus, 2006: 58). Masukan dari plainteks biasanya dibagimenjadi beberapa blok, misalnya 64 bit setiap blok, apabila masukan kurang darijumlah tersebut maka akan dilakukan penambahan bit (padding) sehingga menjadi64 bit. Blok chiper terbagi menjadi empat mode operasi yaitu, Mode Electronic Code Book (ECB), Mode Chiper Blok Chaining (CBC), Mode Chiper Feed Back(CFB), dan Mode Output Feed Back (OFB).

1. Electronic Code Book (ECB)

Ariyus (2006: 58) mengemukakan ECB merupakan "suatu blok chipperyang panjang dibagi dalam bentuk sequence binary menjadi satu blok tanpa mempengaruhi blok-blok yang lain, satu blok terdiri dari 64 bit atau 128 bit, setiap blok merupakan bagian dari pesan yang

dienkripsi". Keunggulan darimode blok ini adalah mode blok chiper yang sederhana, kerusakan satu bloktidak akan mempengaruhi blok yang lainnya. Jika penerima menerima mendapatkan satu blok yang rusak maka blok yang lainnya tidak akan rusakdan penerima hanya perlu dikirim blok yang rusak.

Dengan mode inipengiriman dilakukan dengan cepat karena tidak perlu mengirim ulang pesanyang dikirim secara keseluruhan. Pengiriman blok yang rusak dikirim dengankode yang sama dan menghasilkan chiper yang sama pula. Namun kelemahandari mode ini adalah mudahnya penyerang dalam membaca pola ini karenadapat dipelajari apabila hal ini sering terjadi.Berikut merupakan ilustrasi dari mode ECB yang merupakanpenggambaran dari hal di atas.

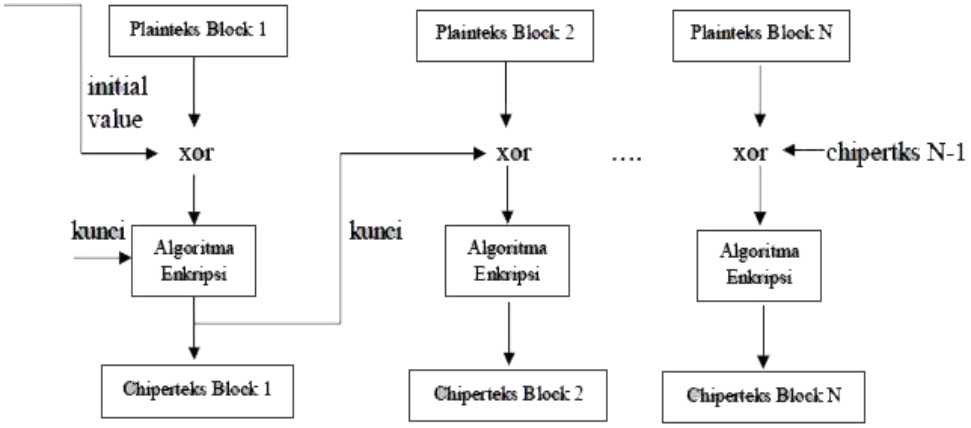


Gambar 2.4 Skema Mode Operasi ECB

(Ariyus, 2006: 58)

2. Chiper Bock Chaining (CBC)

Ariyus (2006: 59) mengemukakan bahwa "sistem dari mode Chiper Bock Chaining (CBC) adalah plainteks yang sama akan dienkripsi ke dalam bentukchiper yang berbeda, disebabkan blok chiper yang satu tidak berhubungandengan blok chiper yang lain. Melainkan tergantung pada chiper yangsebelumnya." Tingkat keamanan dari mode ini lebih rumit dari pada mode ECBdikarenakan tiap blok tergantung dari blok sebelumnya. Namun kalau terjadikesalahan 1 bit saja pada salah satu blok maka akan terjadi kesalahan pula padablok-blok selanjutnya, ini merupakan kelemahan mode CBC. Berikut adalahskema mode CBC.



Gambar 2.5 Skema Mode Operasi CBC

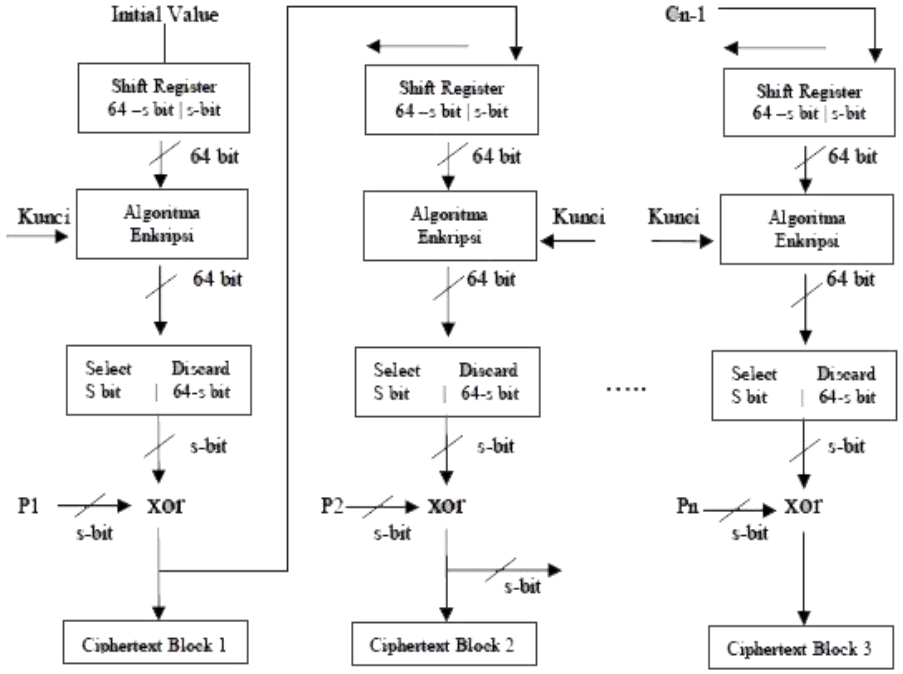
Pada gambar 2.5, terlihat bahwa mode CBC beroperasi menggunakanoperasi XOR antara plainteks dan chiperteks yang berulang. Hasil dari enkripsisatu blok merupakan hasil pengXORan dari blok sebelumnya. IV (initial value)pertama merupakan nilai kunci

asal yang digunakan untuk enkripsi, sedangkan IVpada blok selanjutnya adalah hasil dari blok sebelumnya. Sama halnya denganproses pendekripsian, setiap blok dekripsi merupakan hasil XOR dari blok sebelumnya. IV pada proses dekripsi yang digunakan dari blok sebelumnya laludiXORkan dengan hasil dekripsi. IV merupakan kunci yang harus dilindungikeamanannya oleh pengguna, hal ini dapat dilakukan dengan mode ECB.

3. Cipher Feed Back (CFB)

Metode ini digunakan untuk melakukan enkripsi pada stream chiper, modeini tidak memerlukan padding bit (tambahan bit) karena jumlah panjang bloksama dengan jumlah panjang plainteks yang ada. Mode ini bekerja pada sistemreal time.

Satu hal yang tidak menguntungkan dari mode ini adalah jika satu blokchiper terjadi kesalahan maka, kesalahan untuk semua blok yang lain, karena satu blok dan blok yang lain saling berhubungan. Contoh proses enkripsi darimode CFB diilustrasikan pada gambar di bawah ini.

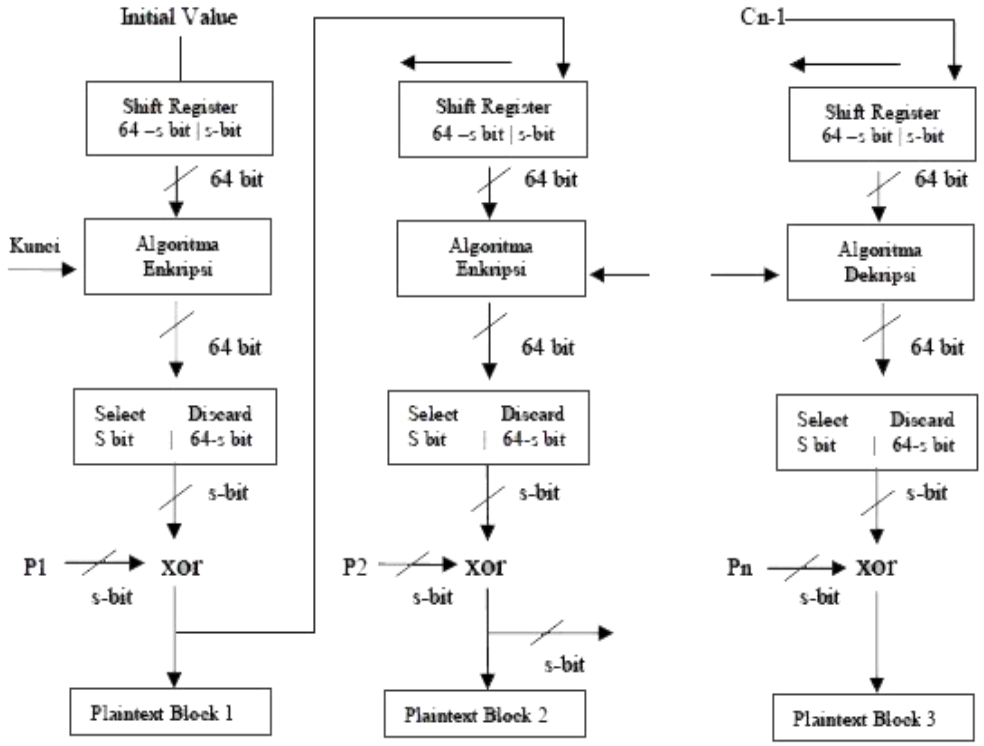


Gambar 2.6 Skema Mode Operasi CFB

Pada gambar di atas s-bit, merupakan pergeseran 1 bit ke kiri dari nilai yang sebenarnya. Hal ini untuk mendapatkan sequence bit yang tidak mudahdilacak oleh penyerang.(Ariyus, 2006: 61)

4. Output Feed Back (OFB)

Ariyus (2006: 62) mengemukakan Mode Output Feed Back (OFB) tidakmempengaruhi blok yang lain jika terjadi error, satu bit yang error padachiperteks hanya akan mempengaruhi satu bit plainteks pada terjadinya prosesdekripsi. Hal ini sangat berguna untuk sistem analog seperti suara atau video, jikasatu bit yang error tidak merusak semua blok yang ada, tapi jika rusak semuamaka akan mempengaruhi arti dari plainteks yang ada .



Gambar 2.7 Skema Mode Operasi OFB

b. Algoritma Stream Cipher

Stream Chiper (aliran chiper) merupakan suatu chiper yang berasal darihasil XOR seperti pembahasan di atas. Setiap bit plainteks dengan setiap bitkunci merupakan kunci utama (kunci induk) yang digunakan untukmembangkitkan kunci acak semu yang dibangkitkan dengan Pseudo-Random Sequence Generator yang merupakan suatu nilai yang nampak seperti diacak,tetapi sesungguhnya nilai tersebut merupakan suatu urutan. Secara khusus urutandari nilai yang dihasilkan oleh RNG (Random Number Generator) merupakankebalikan dari really random.

RNG secara umum adalah Pseudo-random yang memberikan initial stateatau seed (nilai yang diinput ke dalam state), seluruh urutan tersebut ditentukansecara keseluruhan, tetapi meskipun demikian banyaknya

karakteristik yangditampilkan dari suatu urutan yang acak tersebut. Pseudoramdomnessmenghasilkan urutan yang sama secara berulang-ulang pada penempatan yangberbeda. Kemudian kunci acak semu tersebut diberikan operasi XOR denganplainteks untuk mendapatkan chiperteks.(Ariyus, 2006: 52)

2. Synchronous Stream Chiper

Untuk mensimulasikan suatu random, dengan kunci yang mempunyaipanjang terbatas, synchronousstream chiper menghasilkan bit dari sumber yang lain dari pesan itu sendiri. Chiper yang paling sederhana dengan menyadap bitdari register untuk digunakan sebagai kunci. Contentberubah sesuai denganregister yang ada.Suatu struktur yang efisien menghasilkan sequence yang sering digunakanoleh aplikasi random number generator adalah n-element shift register (SR), jika akhir elemen dihubungkan ke elemen yang petama maka nilai n dapat berputarsekitar SR didalam n tetapi jika kedua diantara elemen-elemen dikombinasikanoleh exclusive-OR dan hasilnya dihubungkan ke elemen pertama dan mungkinakan mendapatkan panjang sempurna yang maksimal dari 2n-1 (semua state bernilai nol dan sistem akan mengamankannya).

Karena hanya 2n state yang berbeda dari nilai-nilai n biner dan tiap-tiapnilai dinyatakan tapi cuma bisa digunakan sekali. Nilai-nilai yang dihasilkanadalah suatu permutasi yang sempurna dari perhitungan dari angka-angka (1…2n-1).(Ariyus, 2006: 54)

3. Self-Synchronous Stream Chiper

Self-Synchronous Stream Chiper menggunakan metode pengambilan kuncidari pesan itu sendiri, atau sering disebut dengan autokey chiper, menggunakanpesan untuk kunci. Dengan menggunakan metode ini juga didapat kunci one time pad, kemungkinan setiap pesan dikirim dua kali jarang terjadi, kunci darisistem ini menggunakan pesan yang akan dikirim dengan menambahkan satukarakter yang berbeda di depan maupun di belakang dari kunci yang ada. (Ariyus,2006: 57)

c. 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):

1. Cepat, Blowfish melakukan enkripsi data pada microprocessors 32-bit dengan rate 26 clock cycles perbyte.

2. Compact (ringan), Blowfish dapat dijalankan pada memori kurang dari 5K.

3. Sederhana, Blowfish hanya menggunakan operasi-operasi sederhana: penambahan, XOR, dan lookuptabel pada operan 32-bit.

4. Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh Blowfish dapatbervariasi dan bisa sampai sepanjang 448 bit.

Dalam penerapannya sering kali algortima ini menjadi tidak optimal. Karena strategi implementasi yang tidaktepat. Algoritma Blowfish akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci,seperti jaringan komunikasi atau enkripsi file otomatis. Selain itu, karena algoritma ini membutuhkan memoriyang besar, maka algoritma ini tidak dapat diterapkan untuk aplikasi yang memiliki memori kecil seperti smartcard. Panjang kunci yang digunakan, juga mempengaruhi keamanan penerapan algoritma ini.

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

a. 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, ... Kj $1 \leq j \leq 14$

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

P1, P2, ..., P18

S-box yang terdiri dari 4 buah 32-bit, masing-masing memiliki256 entri :

S1,0, S1,1, ..., S1,255

S2,0, S2,1, ..., S2,255

S3,0, S3,1, ..., S3,255

S4,0, S4,1, ..., S4,255

Langkah-langkah perhitungan atau pembangkitan subkunci tersebut adalah sebagai berikut:

1. Inisialisasi P-array yang pertama dan juga empat S-box,berurutan, dengan string yang telah pasti.String tersebut terdiri dari digit-digit heksadesimal dari phi, tidak termasuk angka tiga di awal.

   Contoh :

   P1= 0x243f6a88

   P2= 0x85a308d3

   P3= 0x13198a2e

   P4= 0x03707344

   dan seterusnya sampai S-box yang terakhir (daftar heksadesimal digit dari phi untuk P-array dan Sboxbisa lihat Lampiran).

2. XOR-kan P1 dengan 32-bit awal kunci, XOR-kan P2 dengan 32-bit berikutnya dari kunci, danseterusnya untuk semua bit kunci. Ulangi siklus seluruh bit kunci secara berurutan

sampai seluruhP-array ter-XOR-kan dengan bit-bit kunci. Atau jika disimbolkan : P1 = P1⊕K1, P2 = P2 ⊕K2, P3 = P3 ⊕ K3, . . . P14 = P14 ⊕K14, P15 = P15 ⊕ K1, . . . P18 = P18 ⊕ K4.

Keterangan : ⊕ adalah simbol untuk XOR.

3. Enkripsikan string yang seluruhnya nol (all-zero string) dengan algoritma Blowfish, menggunakansubkunci yang telah dideskripsikan pada langkah 1 dan 2.
4. Gantikan P1 dan P2 dengan keluaran dari langkah 3.
5. Enkripsikan keluaran langkah 3 menggunakan algoritma Blowfish dengan subkunci yang telahdimodifikasi.
6. Gantikan P3 dan P4 dengan keluaran dari langkah 5.
7. Lanjutkan langkah-langkah di atas, gantikan seluruh elemen P-array dan kemudian keempat S-boxsecara berurutan, dengan hasil keluaran algoritma Blowfish yang terus-menerus berubah.

Total keseluruhan, terdapat 521 iterasi untuk menghasilkan subkunci-subkunci dan membutuhkan memorisebesar 4KB.

b. 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 tambahanlainnya

hanyalah empat penelusuran tabel array berindeks untuk setiap putaran. Langkahnya adalah sepertiberikut.

1. Bagi X menjadi dua bagian yang masing-masing terdiri dari 32-bit: X_L, X_R.
2. Lakukan langkah berikut

   For i = 1 to 16:

   $X_L = X_L \oplus P_i$

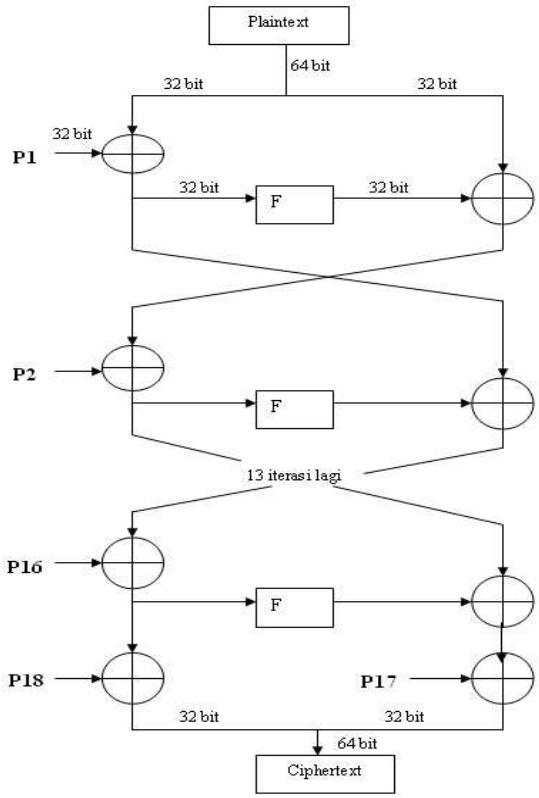
   $X_R = F(X_L) \oplus X_R$

   Tukar X_L dan X_R
3. Setelah iterasi ke-16, tukar X_L dan X_R lagi untuk melakukan membatalkan pertukaran terakhir.
4. Lalu lakukan

   $X_R = X_R \oplus P_{17}$

   $X_L = X_L \oplus P_{18}$
5. Terakhir, gabungkan kembali X_L dan X_R untuk mendapatkan cipherteks.

Untuk lebih jelasnya, gambaran tahapan pada jaringan feistel yangdigunakan Blowfish adalah seperti padaGambar 2.8.

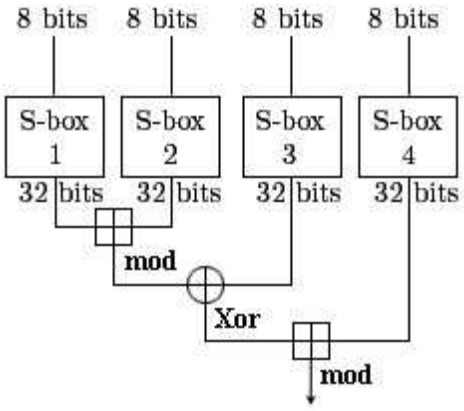


Gambar 2.8 Blok Diagram Algoritma Enkripsi Blowfish

Pada langkah kedua, telah dituliskan mengenai penggunaan fungsi F. Fungsi F adalah: bagi XL menjadi empatbagian 8-bit: a,b,c dan d.

$$F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32} \qquad (2.4)$$

Agar dapat lebih memahami fungsi F, tahapannya dapat dilihat pada Gambar 2.9



Gambar 2.9 Fungsi F dalam Blowfish

Dekripsi sama persis dengan enkripsi, kecuali bahwa P1, P2,…, P18 digunakan pada urutan yang berbalik(reverse). Algoritmanya dapat dinyatakan sebagai berikut (Schneier, 1996) :

for i = 1 to 16 do

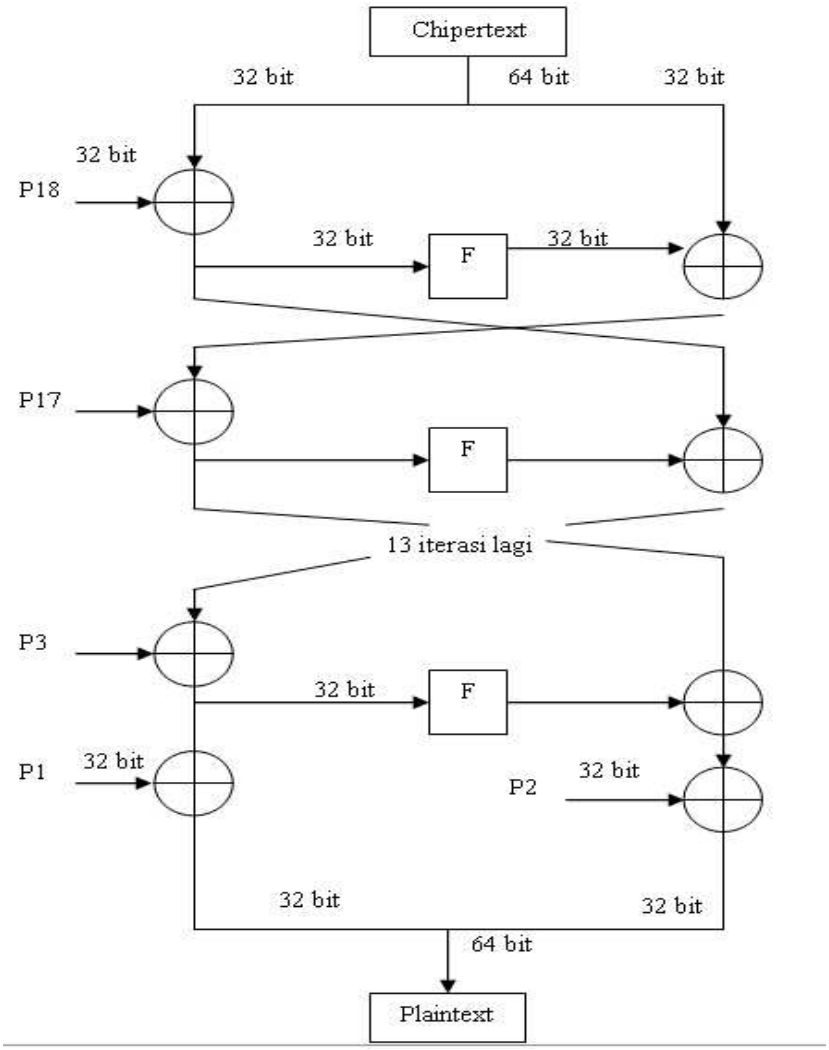
XRi = XLi-1 ⊕ P19-i;

XLi = F[XRi] ⊕ XRi-1;

XL17 = XR16 ⊕ P1;

XR17 = XL16 ⊕ P2;

Blok diagram dekripsi seperti pada Gambar 2.10.



Gambar 2.10 Blok Diagram Dekripsi Blowfish

2.2.6 Pengertian Java

Java adalah suatu teknologi di dunia software komputer, yangmerupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagaibahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkattinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenalC/C++.

Java merupakan bahasa pemrograman berorientasi objek yangmerupakan paradigma pemrograman masa depan. Sebagai bahasapemrograman Java dirancang menjadi handal dan aman. Java juga dirancangagar dapat dijalankan di semua platform. Dan juga dirancang untukmenghasilkan aplikasi – aplikasi dengan performansi yang terbaik, sepertiaplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasapemrograman Java.

Sedangkan Java bersifat neutral architecture, karena JavaCompiler yang digunakan untuk mengkompilasi kode program Java dirancanguntuk menghasilkan kode yang netral terhadap semua arsitekture perangkatkeras yang disebut sebagai Java Bytecode.

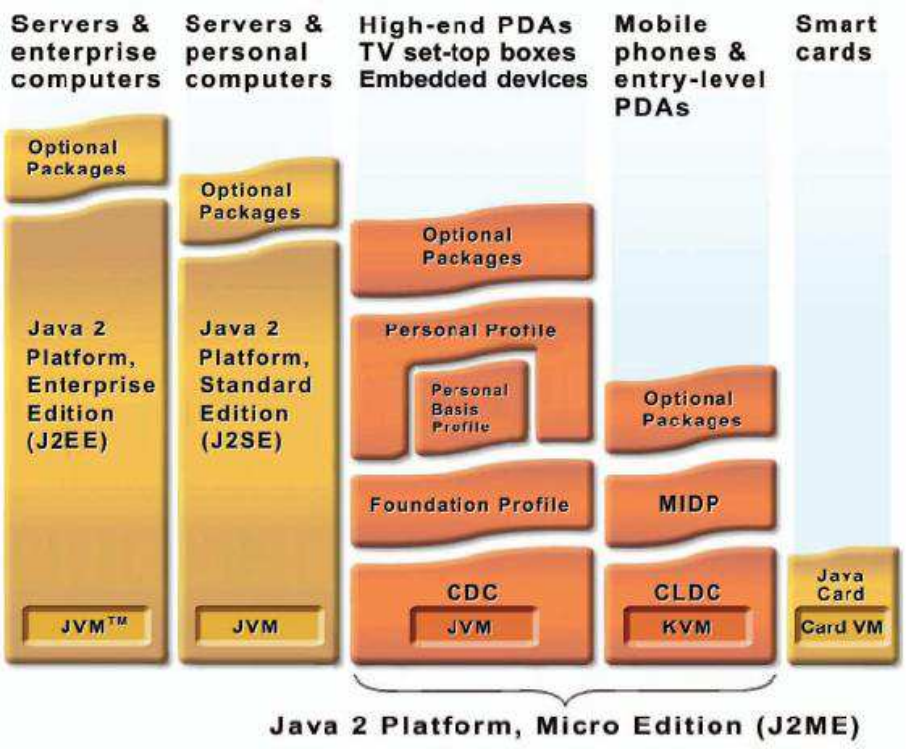
Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu:

- **Java Virtual Machine (JVM).**
- **Java Application Programming Interface (Java API).**

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

- **Enterprise Java (J2EE)** untuk aplikasi berbasis web, aplikasi sistemtersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi.Merupakan superset dari Standar Java

- **Standar Java (J2SE),** ini adalah yang biasa dikenal sebagai bahasa Java.
- **Micro Java (J2ME)** merupakan subset dari J2SE dan salah satuaplikasinya yang banyak dipakai adalah untuk wireless device / mobiledevice.



Gambar 2.11 Java 2 Platform, Micro Edition (J2ME)

a. Sejarah Java

Java diciptakan oleh suatu tim yang dipimpin oleh PatrickNaughton dan James Gosling dalam suatu proyek dari Sun Microsystemyang memiliki kode Green dengan tujuan untuk menghasilkan bahasakomputer sederhana yang dapat dijalankan di peralatan sederhana dengantidak terikat pada arsitekture tertentu. Mulanya disebut OAK, tetapikarena OAK sendiri merupakan nama dari bahasa pemrograman computer yang sudah ada. Maka Sun mengubahnya menjadi Java.

Sun kemudian meluncurkan browser dari Java yang disebut HotJava yang mampu menjalankan applet. Setelah itu teknologi Java diadopsioleh Netscape yang memungkinkan program Java dijalankan di browserNetscape yang kemudian diikuti Internet Explorer. Karena keunikanyadan kelebihanya, teknologi Java mulai menarik banyak vendor sepertiIBM,Symantec, Inprise, dll.

Sun merilis versi awal Java secara resmi pada awal tahun 1996yang kemudian terus berkembang hingga muncul JDK 1.1, kemudian JDK1.2 yang mulai disebut sebagai versi Java2 karena banyak mengandungpeningkatan dan perbaikan. Perubahan utama adalah adanyaSwing yangmerupakan teknologi GUI (Graphical User Interface) yang mampumenghasilkan window yang portabel. Dan pada tahun 1998 – 1999lahirlah teknologi J2EE (Java 2 Enterprise Edition) yang berbasis J2SEyang diawali dengan servlet dan EJB kemudian diikuti JSP. Java jugamenjadi lebih cepat populer di lingkungan server side

dikarenakankelebihanya di lingkungan network dan terdistribusi serta kemampuanmultithreading.

Sedangkan J2ME (Java 2 Micro Edition) dapatmenghasilkan aplikasi mobile baik games maupun software yang dapatdijalankan di peralatan mobile seperti ponsel.

b. Fitur – Fitur Java yang Menarik

Beberapa fitur yang ditawarkan Java API antara lain sebagaiberikut :

1. Applet

Program Java yang dapat berjalan di atas browser, yang dapatmembuat halaman HTML lebih dinamis dan menarik.

2. Java Networking

Sekumpulan API (Application Programming Interface) yangmenyediakan fungsi – fungsi untuk aplikasi – aplikasi jaringan,seperti penyediaan akses untuk TCP, UDP, IP Adrress dan URL.Tetapi Java Networking tidak menyediakan akses untuk ICMPdikarenakan alasan sekuriti dan pada kondisi umum hanyaadministrator (root) yang bisa memanfaatkan protokol ICMP.

3. Java Database Connectivity (JDBC)

JDBC menyediakan sekumpulan API yang dapat digunakan untukmengakses database seperti Oracle, MySQL, PostgreSQL,Microsoft SQL Server.

4. Java Security

Java Security menyediakan sekumpulan API untuk mengatursecurity dari aplikasi Java baik secara high level atau low level,seperti public/private key management dan certificates.

5. Java Swing

Java Swing menyediakan sekumpulan API untuk membangunaplikasi – aplikasi GUI (Graphical User Interface) dan model GUIyang diinginkan bisa bermacam – macam, bisa model Java, modelMotif/CDE atau model yang dependent terhadap platform yangdigunakan.

6. JavaRemote Method Invocation(RMI)

Java RMI menyediakan sekumpulan API untuk membangunaplikasi – aplikasi Java yang mirip dengan model RPC (Remote Procedure Call) jadi object - object Java bisa di call secara remotepada jaringan komputer.

7. Java 2D/3D

Java 2D/3D menyediakan sekumpulan API untuk membangungrafik – grafik 2D/3D yang menarik dan juga akses ke printer.

8. Java Server Pages

Berkembang dari Java Servlet yang digunakan untukmenggantikan aplikasi – aplikasi CGI, JSP (Java Server Pages)yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusiaplikasi Internet.

9. Java Native Interface(JNI)

JNI menyediakan sekumpulan API yang digunakan untukmengakses fungsi – fungsi pada library (*.dll atau *.so)

yangdibuat dengan bahasa pemrograman yang lain seperti C,C++, danBasic.

10. Java Sound

Java Sound menyediakan sekumpulan API untuk manipulasisound.

11. Java IDL + CORBA

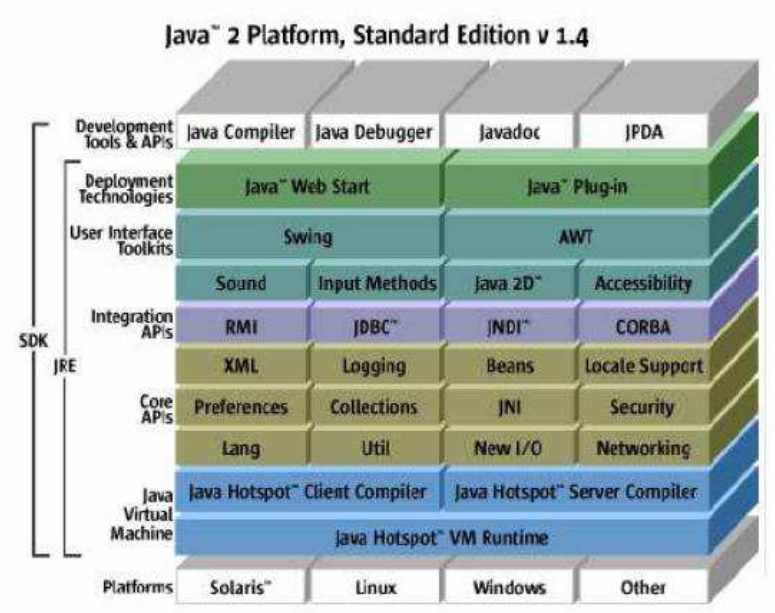
Java IDL (Interface Definition Language) menyediakan dukunganJava untuk implementasi CORBA (Common Object RequestBroker) yang merupakan model distributed-Object untuk solusiaplikasi besar di dunia networking.

12. Java Card

Java Card utamanya digunakan untuk aplikasi – aplikasi padasmart card, yang sederhana wujudnya seperti SIM Card padahandphone.

13. Java Telephony API(JTAPI)

Java Telepony API menyediakan sekumpulan API untukmemanfaatkan devices – devices telepony, sehingga akan cocokuntuk aplikasi – aplikasi CTI (Computer Telephony Integration)yang dibutuhkan seperti ACD (Automatic Call Distribution), PCPBXdan lainnya.



Gambar 2.12 Java™ 2 Platform, Standar Edition v 1.4

2.2.7 Pengertian File

File merupakan dokumen yang mengandung informasi tertentu dandapat dibuka dengan program. (Hakim S, Rachmad. 2010)

File extension adalah akhiran yang mengikuti sebuah file, selalu dipisahkan dengan tanda titik atau dot (.), misalnya file.png, dimana png adalah extension dari file. Extension dari file ini membantu sistem operasi seperti windows untuk menentukan aplikasi atau program yang sesuai dengan file tersebut. Beberapa jenis file extension antara lain:

Text Files Types and Formats

.doc Microsoft Word Document

.docx Microsoft Word Open XML Document

.log Log File

.msg Outlook Mail Message

.pages Pages Document

.rtf Rich Text Format File

.txt Plain Text File

.wpd WordPerfect Document

.wps Microsoft Works Word Processor Document

Data Files Types and Formats

.csv Comma Separated Values File

.dat Data File

.efx eFax Document

.gbr Gerber File

.key Keynote Presentation

.pps PowerPoint Slide Show

.ppt PowerPoint Presentation

.pptx PowerPoint Open XML Presentation

.sdf Standard Data File

.tax2010 TurboTax 2010 Tax Return

.vcf vCard File

.xml XML File

Audio File Types and Formats

.aif Audio Interchange File Format

.iff Interchange File Format

.m3u Media Playlist File

.m4a MPEG-4 Audio File

.mid MIDI File

.mp3 MP3 Audio File

.mpa MPEG-2 Audio File

.ra Real Audio File

.wav WAVE Audio File

.wma Windows Media Audio File

Video Files Types and Formats

.3g2 3GPP2 Multimedia File

.3gp 3GPP Multimedia File

.asf Advanced Systems Format File

.asx Microsoft ASF Redirector File

.avi Audio Video Interleave File

.flv Flash Video File

.mov Apple QuickTime Movie

.mp4 MPEG-4 Video File

.mpg MPEG Video File

.rm Real Media File

.swf Shockwave Flash Movie

.vob DVD Video Object File

.wmv Windows Media Video File

3D Image Files Types and Formats

.3dm Rhino 3D Model

.max 3ds Max Scene File

Raster Image Files Types and Formats

.bmp Bitmap Image File

.gif Graphical Interchange Format File

.jpg JPEG Image File

.png Portable Network Graphic

.psd Adobe Photoshop Document

.pspimage PaintShop Pro Image

.thm Thumbnail Image File

.tif Tagged Image File

.yuv YUV Encoded Image File

Vector Image Files Types and Formats

.ai Adobe Illustrator File

.drw Drawing File

.eps Encapsulated PostScript File

.ps PostScript File

.svg Scalable Vector Graphics File

Page Layout Files Types and Formats

.indd Adobe InDesign Document

.pct Picture File

.pdf Portable Document Format File

.qxd QuarkXPress Document

.qxp QuarkXPress Project File

.rels Open Office XML Relationships File

Spreadsheet Files Types and Formats

.xlr Works Spreadsheet

.xls Excel Spreadsheet

.xlsx Microsoft Excel Open XML Spreadsheet

Database Files Types and Formats

.accdb Access 2007 Database File

.db Database File

.dbf Database File

.mdb Microsoft Access Database

.pdb Program Database

.sql Structured Query Language Data

Executable Files Types and Formats

.app Mac OS X Application

.bat DOS Batch File

.cgi Common Gateway Interface Script

.com DOS Command File

.exe Windows Executable File

.gadget Windows Gadget

.jar Java Archive File

.pif Program Information File

.vb VBScript File

.wsf Windows Script File

Game Files Types and Formats

.gam Saved Game File

.nes Nintendo (NES) ROM File

.rom N64 Game ROM File

.sav Saved Game

CAD Files Types and Formats

.dwg AutoCAD Drawing Database File

.dxf Drawing Exchange Format File

GIS Files Types and Formats

.gpx GPS Exchange File

.kml Keyhole Markup Language File

Web Files Types and Formats

.asp Active Server Page

.cer Internet Security Certificate

.csr Certificate Signing Request File

.css Cascading Style Sheet

.htm Hypertext Markup Language File

.html Hypertext Markup Language File

.js JavaScript File

.jsp Java Server Page

.php Hypertext Preprocessor File

.rss Rich Site Summary

.xhtml Extensible Hypertext Markup Language File

Plugin Files Types and Formats

.8bi Photoshop Plug-in

.plugin Mac OS X Plug-in

.xll Excel Add-In File

Font Files Types and Formats

.fnt Windows Font File

.fon Generic Font File

.otf OpenType Font

.ttf TrueType Font

System Files Types and Formats

.cab Windows Cabinet File

.cpl Windows Control Panel Item

.cur Windows Cursor

.dll Dynamic Link Library

.dmp Windows Memory Dump

.drv Device Driver

.lnk File Shortcut

.sys Windows System File

Settings Files Types and Formats

.cfg Configuration File

.ini Windows Initialization File

.keychain Mac OS X Keychain File

.prf Outlook Profile File

Encoded Files Types and Formats

.bin Macbinary Encoded File

.hqx BinHex 4.0 Encoded File

.mim Multi-Purpose Internet Mail Message File

.uue Uuencoded File

Compressed Files Types and Formats

.7z 7-Zip Compressed File

.deb Debian Software Package

.gz Gnu Zipped Archive

.pkg Mac OS X Installer Package

.rar WinRAR Compressed Archive

.rpm Red Hat Package Manager File

.sit StuffIt Archive

.sitx StuffIt X Archive

.tar.gz Tarball File

.zip Zipped File

.zipx Extended Zip File

Disk Image Files Types and Formats

.dmg Mac OS X Disk Image

.iso Disc Image File

.toast Toast Disc Image

.vcd Virtual CD

Developer Files Types and Formats

.c C/C++ Source Code File

.class Java Class File

.cpp C++ Source Code File

.cs Visual C# Source Code File

.dtd Document Type Definition File

.fla Adobe Flash Animation

.java Java Source Code File

.m Objective-C Implementation File

.pl Perl Script

.py Python Script

Backup Files Types and Formats

.bak Backup File

.gho Norton Ghost Backup File

.ori Original File

.tmp Temporary File

Misc Files Types and Formats

.dbx Outlook Express E-mail Folder

.msi Windows Installer Package

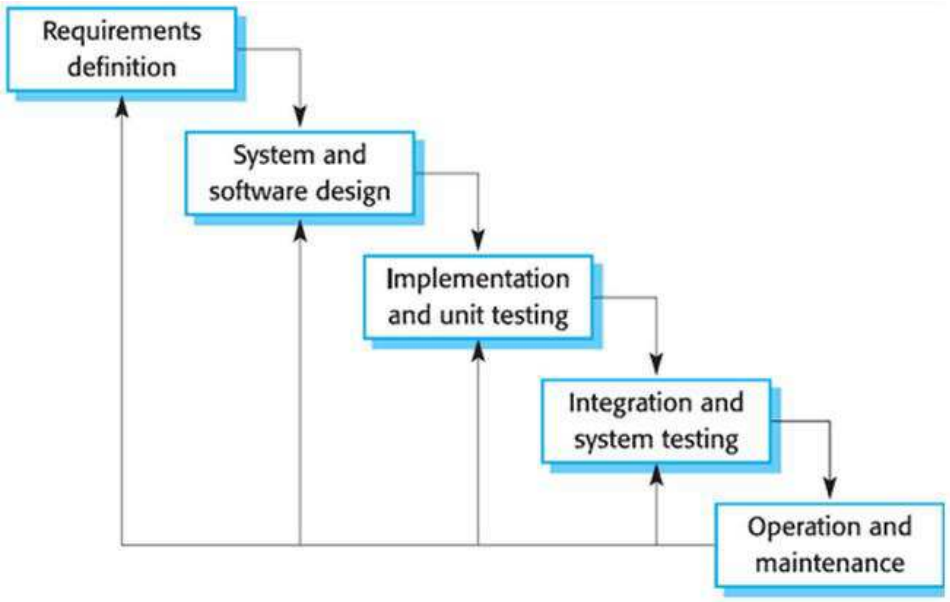
.part Partially Downloaded File

.torrent BitTorrent File

2.2.8 Metode Waterfall

Model pengembangan perangkat lunak yang diperkenalkan oleh Winston Royce pada tahun 70-an ini merupakan model klasik yang sederhana dengan aliran sistem yang linier — keluaran dari tahap sebelumnya merupakan masukan untuk tahap berikutnya. Pengembangan dengan model ini adalah hasil adaptasi dari pengembangan perangkat keras, karena pada waktu itu belum terdapat metodologi pengembangan perangkat lunak yang lain. Proses pengembangan yang sangat terstruktur ini membuat potensi kerugian akibat kesalahan pada proses sebelumnya sangat besar dan acap kali mahal karena membengkaknya biaya pengembangan ulang.

Metode Waterfall adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Berikut adalah gambar pengembangan perangkat lunak berurutan/ linear (Pressman, Roger S. 2001):



Gambar 2.13 Metode Waterfall

a. Tahapan Metode Waterfall

Dalam pengembangannya metode waterfall memiliki beberapa tahapan yang runtut: requirement (analisis kebutuhan), design sistem (system design), Coding & Testing, penerapan program, pemeliharaan.

1. Requirement (analisis kebutuhan)

Dalam langakah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur.

Seseorang system analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan system analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. System Design (desain sistem)

Proses design akan menterjemahkan syarat kebutuhan kesebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. Coding & Testing (penulisan sinkode program / implementation)

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer.Dilakukan oleh programmer yang akan meterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer

akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap system tersebut dan kemudian bisa diperbaiki.

4. Integration & Testing (Penerapan / Pengujian Program)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadikan digunakan oleh user.

5. Operation & Maintenance (Pemeliharaan)

Perangkat lunak yang susah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

b. Manfaat Metode Waterfall

Keunggulan model pendekatan pengembangan perangkat lunak dengan metode waterfall adalah pencerminan kepraktisan rekayasa, yang membuat kualitas perangkat lunak tetap terjaga karena pengembangannya yang terstruktur dan terawasi. Disisi lain model ini merupakan jenis model yang bersifat dokumen lengkap, sehingga proses pemeliharaan dapat dilakukan dengan mudah. Akan tetapi

dikarenakan dokumentasi yang lengkap dan sangat teknis, membuat pihak klien sulit membaca dokumen yang berujung pada sulitnya komunikasi antar pengembang dan klien.

Dokumentasi kode program yang lengkap juga secara tak langsung menghapus ketergantungan pengembang terhadap pemrogram yang keluar dari tim pengembang. Hal ini sangat menguntungkan bagi pihak pengembang dikarenakan proses pengembangan perangkat lunak tetap dapat dilanjutkan tanpa bergantung pada pemrogram tertentu.

c. Kelemahan Metode Waterfall

Kelemahan pengembangan perangkat lunak dengan metode waterfall yang utama adalah lambatnya proses pengembangan perangkat lunak. Dikarenakan prosesnya yang satu persatu dan tidak bisa diloncat-loncat menjadikan model klasik ini sangat memakan waktu dalam pengembangannya. Disisi lain, pihak klien tidak dapat mencoba sistem sebelum sistem benar-benar selesai pembuatannya. Kelemahan yang lain adalah kinerja personil yang tidak optimal dan efisien karena terdapat proses menunggu suatu tahapan selesai terlebih dahulu.

BAB III

METODE PENELITIAN

3.1 Prosedur Penelitian

Untuk dapat melaksankan tahapan penelitian maka tahapan penelitian yang dilakukan adalah sebagai berikut:

3.1.1 Studi Pustaka.

Studi pustaka dilakukan untuk mencari informasi - informasi tentang teori, metode dan konsep yang relevan dengan permasalahan. Sehingga dengan informasi – informasi tersebut dapat digunakan sebagai acuan dalam penyelesaian masalah. Studi pustaka yang dilakukan dengan mencari informasi dan referensi dalam bentuk text book, literatur, informasi dari internet maupun sumber-sumber lainnya yang berkaitan dengan penelitian ini.

3.1.2 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Tujuan yang diungkapkan dalam bentuk hipotesis merupakan jawaban sementara terhadap pertanyaan penelitian.metode pengumpulan data bisa dilakukan dengan cara:

a. Observasi

   Merupakan cara untuk mendapatkan data dan informasi dengan melakukan peninjauan atau pengamatan secara

langsung ketempat yang berkaitan dengan penulisan tugas akhir dan perancangan sistem informasinya.

b. Komparatif

Merupakan cara untuk mendapatkan data dan informasi dengan melakukan pertimbangan atau membandingkan dua sistem atau lebih, melihat kekurangan dan kelebihan antara sistem yang lama dengan sistem yang baru. Sehingga dapat dilakukan pengembangan sistem yang baru.

3.1.3 Metode Penulisan

Metodologi yang digunakan dalam pembangunan aplikasi ini adalah :

a. Metode Studi Pustaka

Metode Studi Pustaka ini dilakungan dengan cara mengumpulkan dan mempelajari teori - teori literatur dari berberapa buku referensi, skripsi, jurnal serta data- data penunjang lainnya yang berhubungan dengan judul skripsi penulis dalam menyelesaikan masalah.

b. Metode Pengembangan Perangkat Lunak

1) Analisis

Analisis dilakungan dengan cara menganalisa permasalahan yang muncul dan menentukan spesifikasi kebutuhan atas sistem yang akan dibuat. Hasil dari analisa berupa Dokumen Spesifikasi Kebutuhan Prangkat Lunak (SKPL).

2) Perancangan

Perancangan sistem dilakukan dengan berdasarkan hasil analisis yang telah dilakukan pada tahap sebelumnya. Perancangan dilakukan untuk mendapatkan deskripsi arsitektural perangkat lunak, deskripsi data dan deskripsi prosedural. Metode yang digunakan pada perancangan aplikasi kriptografi fileadalah metode waterfall.

3) Pengkodean

Pengkodean dilakukan dengan mengimplentasi hasil analisis dan perancangan kedalam program untuk membangun aplikasi. Hasil dari pengkodean adalah kode yang siap di eksekusi.

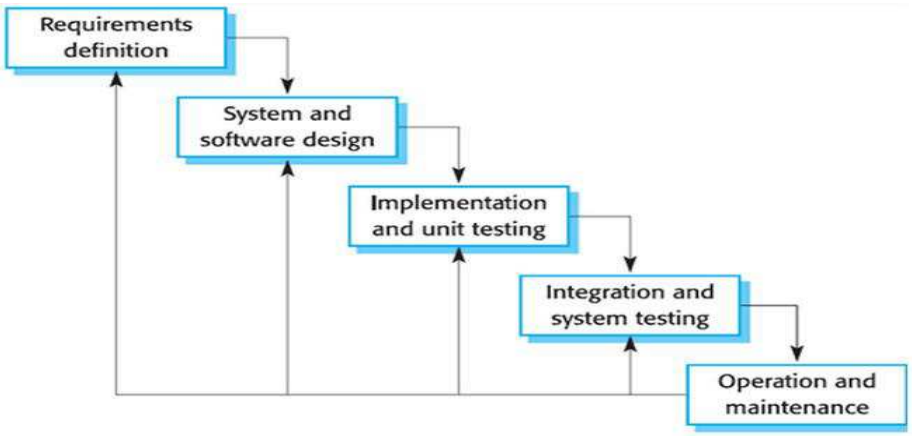
4) Pengujian

Pengujian dialakukan dengan menguji aplikasi yang sudah dibuat pada langkah pengkodean, pengujian ini meliputi pengujian fungsional, dan pengujian hasil apakah sudah sesuai dengan kebutuhan dalam dokumen perancangan.

3.2. Metode Waterfall

Model pengembangan perangkat lunak yang diperkenalkan oleh Winston Royce pada tahun 70-an ini merupakan model klasik yang sederhana dengan aliran sistem yang linier — keluaran dari tahap sebelumnya merupakan masukan untuk tahap berikutnya. Pengembangan dengan model ini adalah hasil adaptasi dari pengembangan perangkat keras, karena pada waktu itu belum terdapat metodologi pengembangan perangkat lunak yang lain. Proses pengembangan yang sangat terstruktur ini membuat potensi kerugian akibat kesalahan pada proses sebelumnya sangat besar dan acap kali mahal karena membengkaknya biaya pengembangan ulang.

Metode Waterfall adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Berikut adalah gambar pengembangan perangkat lunak berurutan/ linear (Pressman, Roger S. 2001):



Gambar 3.1 Metode Waterfall

a. Tahapan Metode Waterfall

Dalam pengembangannya metode waterfall memiliki beberapa tahapan yang runtut: requirement (analisis kebutuhan), design sistem (system design), Coding & Testing, penerapan program, pemeliharaan.

1. Requirement (analisis kebutuhan)

Dalam langakah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur. Seseorang system analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan system analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. System Design (desain sistem)

Proses design akan menterjemahkan syarat kebutuhan kesebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement. Dokumen inilah

yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. Coding & Testing (penulisan sinkode program / implementation)

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer.Dilakukan oleh programmer yang akan meterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap system tersebut dan kemudian bisa diperbaiki.

4. Integration & Testing (Penerapan / Pengujian Program)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadikan digunakan oleh user.

5. Operation & Maintenance (Pemeliharaan)

Perangkat lunak yang susah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi

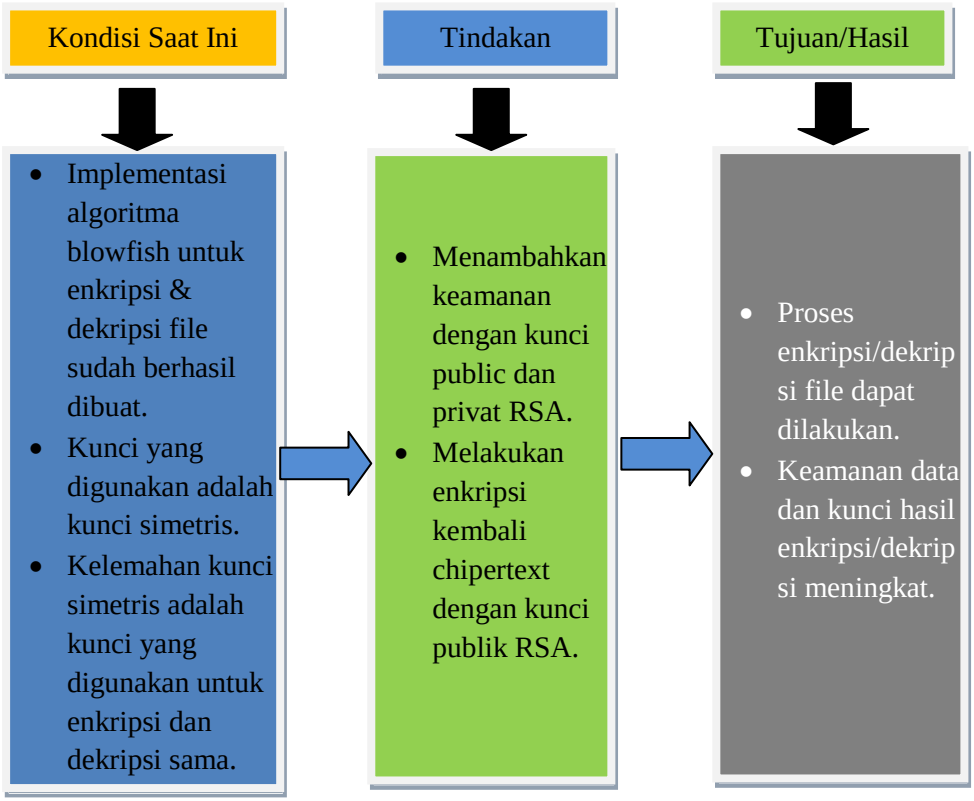
baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

3.3. Kerangka Berpikir

Algoritma blowfish adalah algoritma kunci simetris, dimana kunci untuk enkripsi dan dekripsi sama. Keamanannya terletak pada panjang kunci yang digunakan dan kerahasiaan kunci itu sendiri.

Algoritma RSA adalah algoritma kunci asimetris, dimana kunci untuk enkripsi (kunci publik) berbeda dengan kunci untuk dekripsi (kunci privat). Kunci publik tidak rahasia sedangkan kunci privat bersifat rahasia.

Dengan menggabungkan algoritma blowfish dengan algoritma RSA pada aplikasi enkripsi file diharapkan keamanan dari file tersebut semakin meningkat.



Gambar 3.2Kerangka Berpikir

BAB IV

PERANCANGAN SISTEM

4.1. Analisis Sistem

Masalah yang diselesaikan dalam penulisan ini antara lain adalahmenerapkan teknik kunci asimetris pada algoritma blowfish digunakan untuk enkripsi dan dekripsi file. Pada subbab ini dilakukan beberapa analisis yaitu deskripsi sistem danperancangan proses sistem yang akan dibangun.

4.2. Deskripsi Sistem

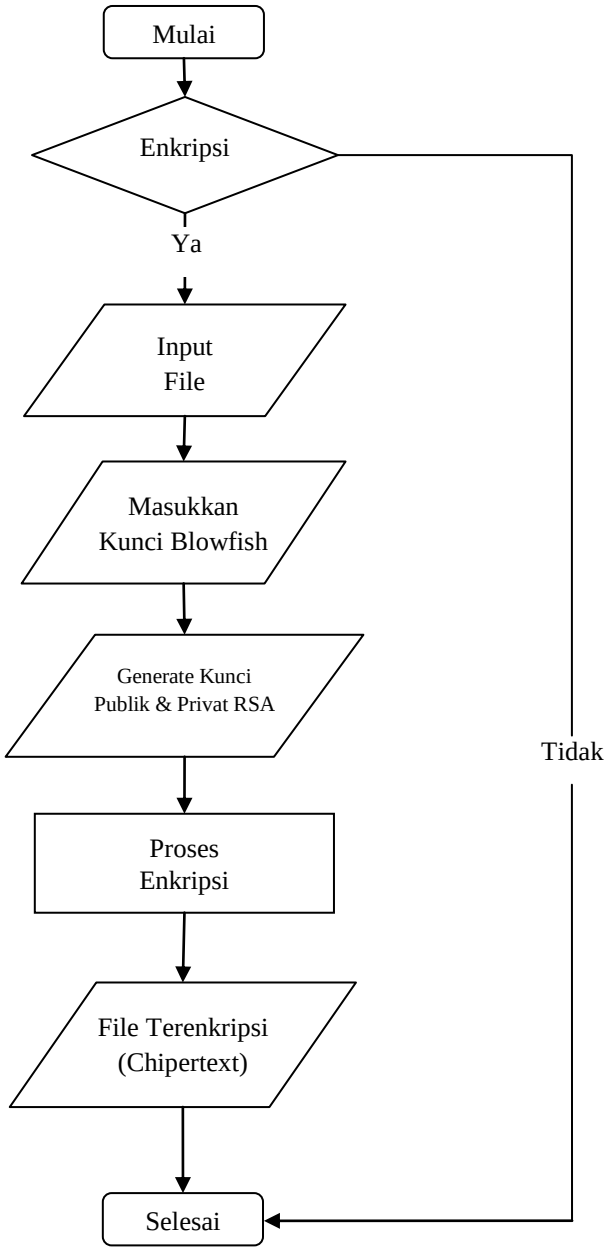
Subbab ini akan membahas mengenai deskripsi sistem yang dikerjakanpada skripsi ini. Tujuan pembuatan sistem ini adalah menerapkan algoritma untukmengamankan file sehingga file tersebut menjadi tidakdapat terbaca. Proses utama pada aplikasi perangkat lunak ini adalah melakukanenkripsi dan dekripsi. Teknik kunci asimetris dengan algoritma RSA digunakan untuk mengenkripsi kunci simetris algoritma blowfish sehingga keamanan enkripsi dan dekripsi meningkat. Adapun proses dalam perangkat lunak ini sebagai berikut :

a. Melakukan enkripsi file.

1. Pengguna memasukkan input berupa file. File yang akandiinputkan berupa file teks.
2. Masukkan kunci blowfishuntuk mengenkripsi.
3. Lakukan enkripsi fileyang telah diinputkan.

4. Fileyang telah terenkripsi menjadi file yang tidakterbaca (chipertext).
5. Chipertext dienkripsi kembali dengan algoritma RSA.
6. RSA kunci generator (kunci publik dan kunci privat).
7. Enkripsi selesai.

Diagram alir untuk enkripsi file adalah sebagai berikut:



Gambar 4.1 Enkripsi File

b. Melakukan dekripsi file

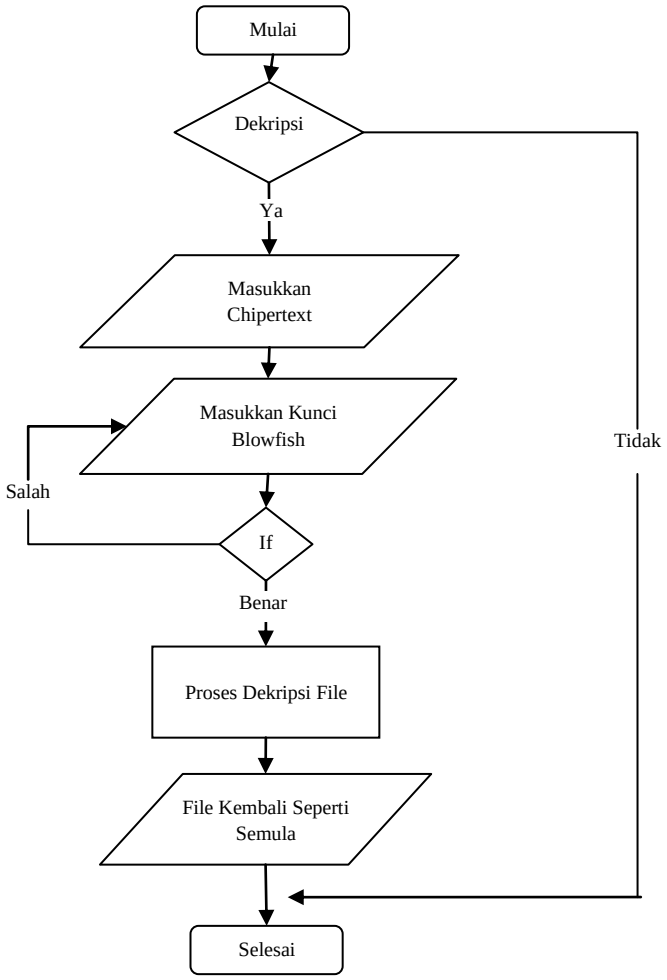
1. Masukkan chipertext.

2.Masukkan kunci blowfish.

3. Lakukan dekripsi file.

4. Selesai.

Diagram alir dekripsi file adalah sebagai berikut :



Gambar 4.2 Dekripsi File

c. 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.

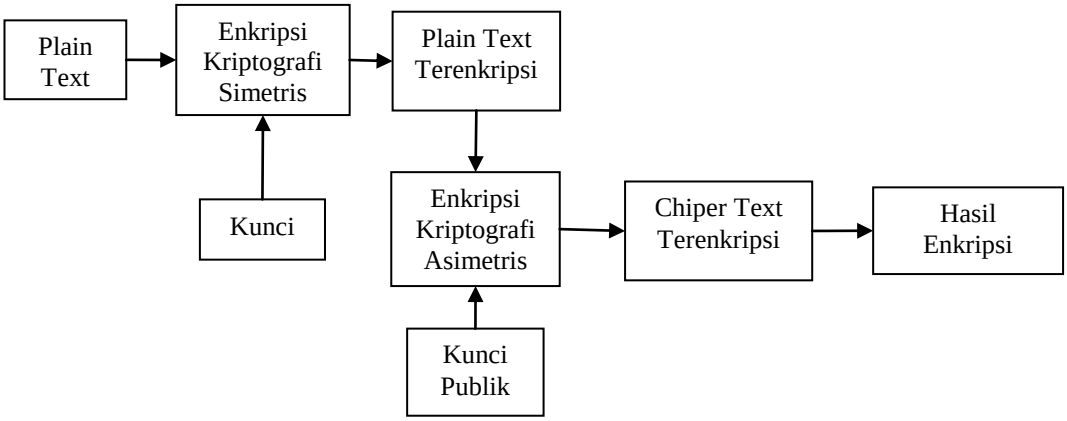
   • n disebut juga modulus

   • e disebut juga public exponent atau exponentenkripsi

   • d disebut juga secret exponent atau exponentdekripsi

d. Melakukan enkripsi kunci simetris dan plain text terenkripsi dengan enkripsi kriptografi asimetris RSA

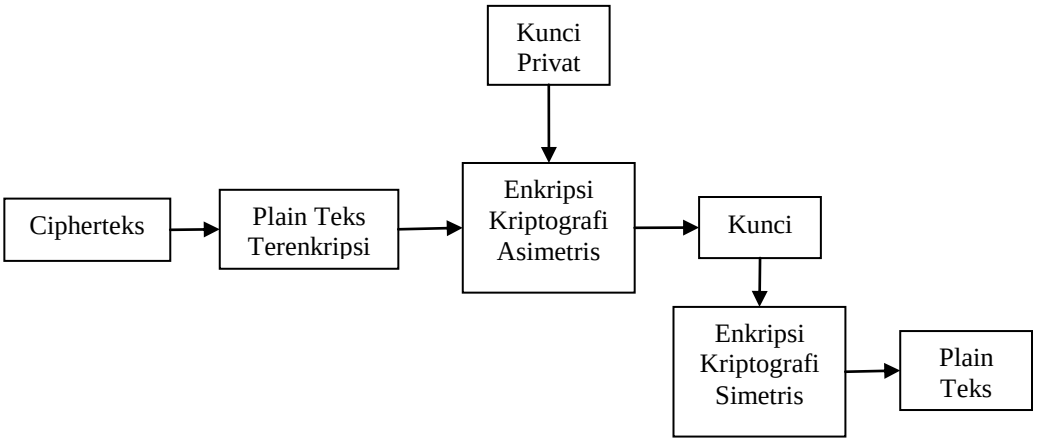
1. Ambil kunci public (n,e)
2. Nyatakan plainteks dalam integer positif m
3. Enkripsi menjadi cipher teks $c = m^e \bmod n$



Gambar 4.3 Enkripsi Dengan Kunci Publik

e. Melakukan dekripsi kunci simetris dan plainteks terenkripsi dengan dekripsi kriptografi asimetris RSA.

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



Gambar 4.4 Dekripsi Dengan Kunci Privat

4.3 Perancangan Aplikasi Enkripsi File

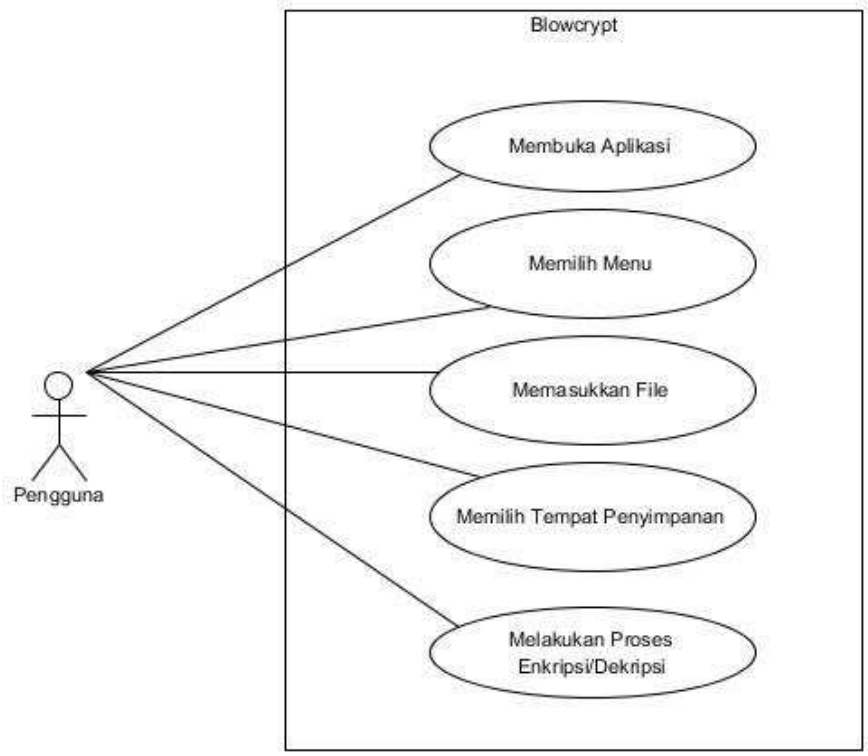
Aplikasi ini akan diberi nama asimetrisblowcrypt v1.0merupakan program yang berbasis objek, untuk mempresentasikan jalannya program makan dibutuhkan beberapa metode perancangan sistem.

4.3.1 Perancangan Menggunakan Unified Modeling Language (UML)

Diagram UML yang akan dipaparkan oleh penulis yaitu Use Case Diagramdan Activity Diagram.

a. Use Case Diagram

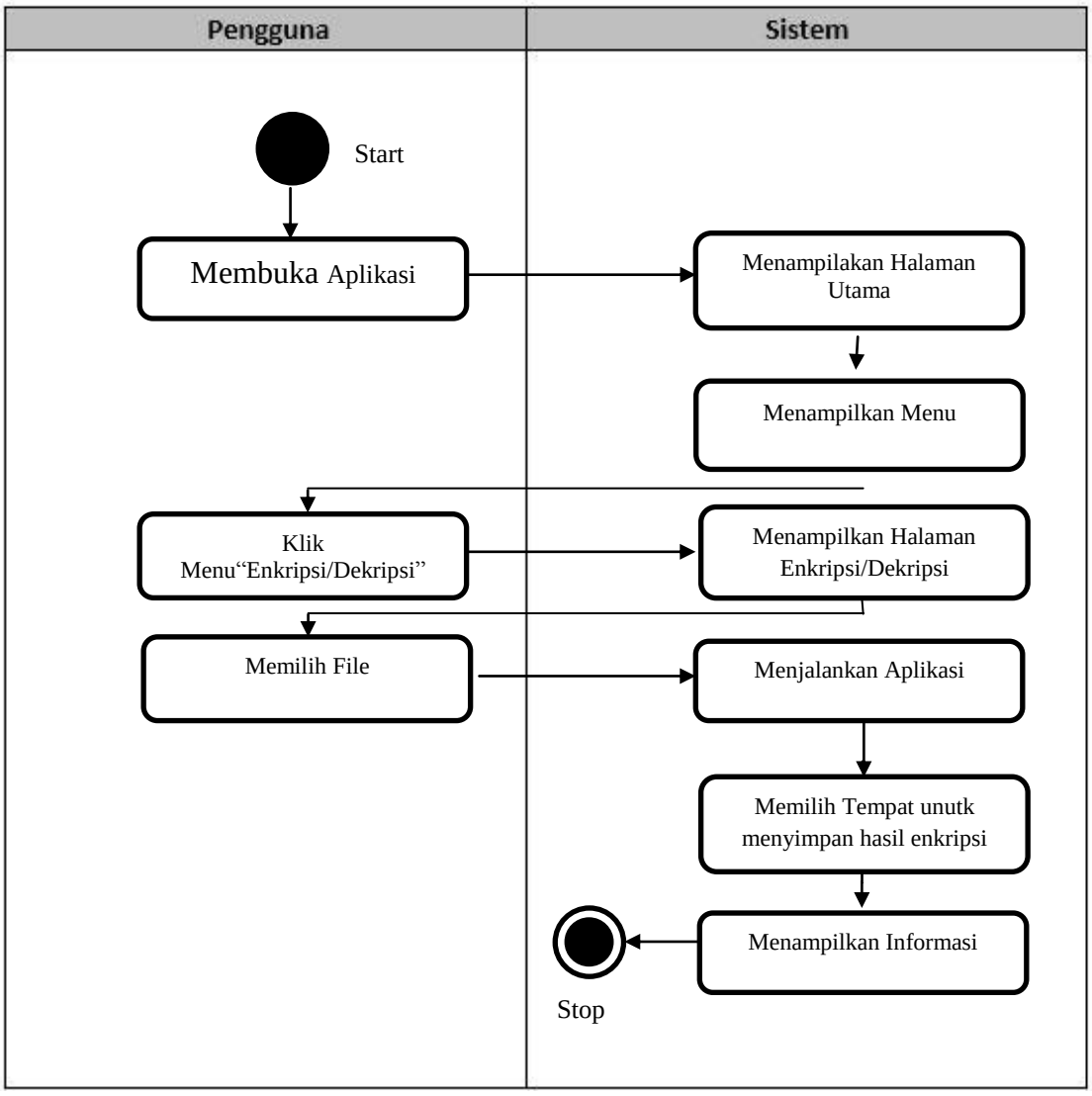
Diagram Use Case ini menggambarkan pengguna yang akan menggunakan sistem dan prilaku pengguna terhadap sistem sebagai aktor yang terlibat dalam sistem.



Gambar 4.5Use Case Diagram.

b. Activity Diagram

Activitydiagrammenguraikan sistem yang akan berjalan, Gambar 3.4 berikut menunjukan Activity diagram pada sistem asimetrisblowcrypt v1.0.



Gambar 4.6Activity Diagram.

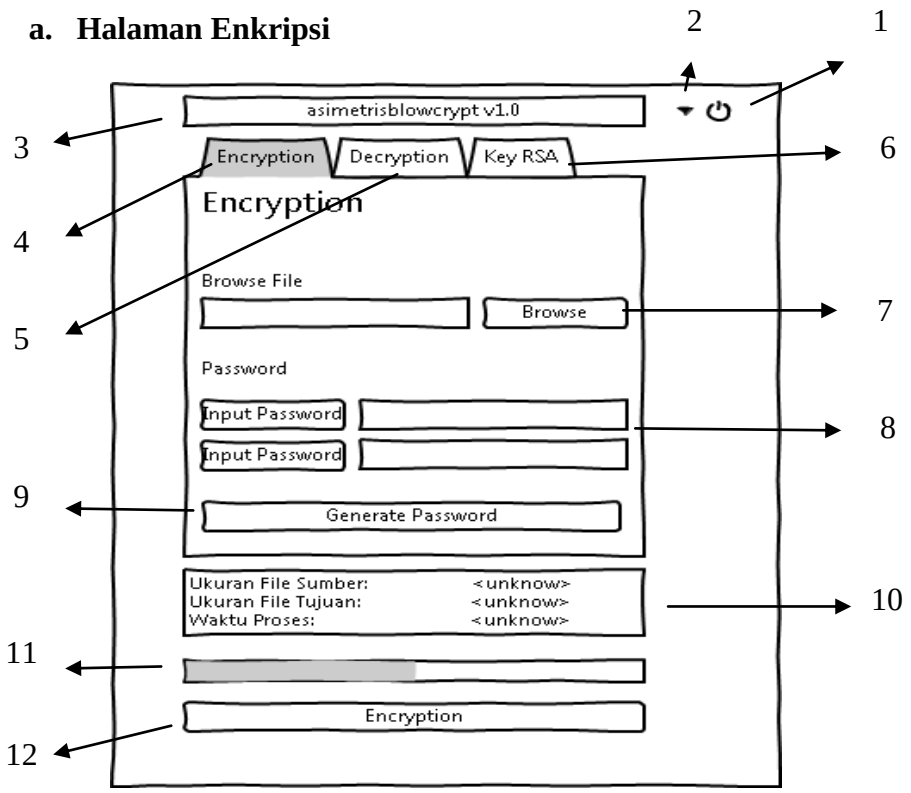
Pada gambar 3.6 dapat dijelaskan bahwa langkah penggunaan aplikasi oleh pengguna, dimulai ketika pengguna membuka aplikasi kriptografi file dengan algoritma blowfishdan memunculkan halaman utama, langkah – langkah diatas dapat diuraikan sebagai berikut:

1. Membuka Aplikasi asimetrisblowcrypt v1.0.
2. Sistem akan menampilkan halaman utama dari aplikasi.
3. Pengguna bisa memilih beberapa menu pada aplikasi, antara lain: enkripsi/dekripsi, key RSA, dan keluar.
4. Ketika pengguna mengklik menu enkripsi/dekripsi, maka akan muncul halaman enkripsi/dekripsi.
5. Untuk memasukkan file yang akan di enkripsi/dekripsi pengguna harus menekan tombol "Browse".
6. Untuk memulai proses enkripsi/dekripsi pengguna harus menekan tombol "Encryption".
7. Jika enkripsi/dekripsi file berhasil akanada notifikasi "enkripsi/dekripsi sukses"
8. Untuk menutup aplikasi dengan menekan tombol "Tutup".

4.3.2 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.

a. Halaman Enkripsi



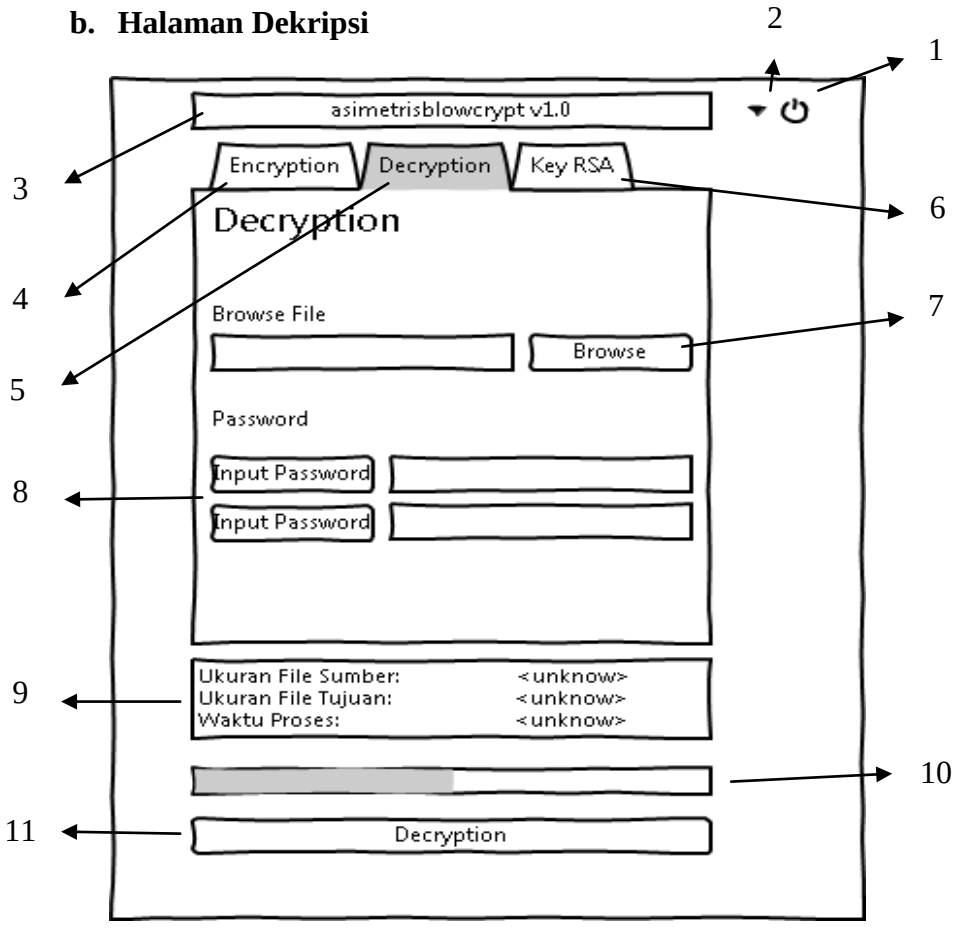
Gambar 4.7 Tampilan Antarmuka Halaman Enkripsi

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.

b. Halaman Dekripsi



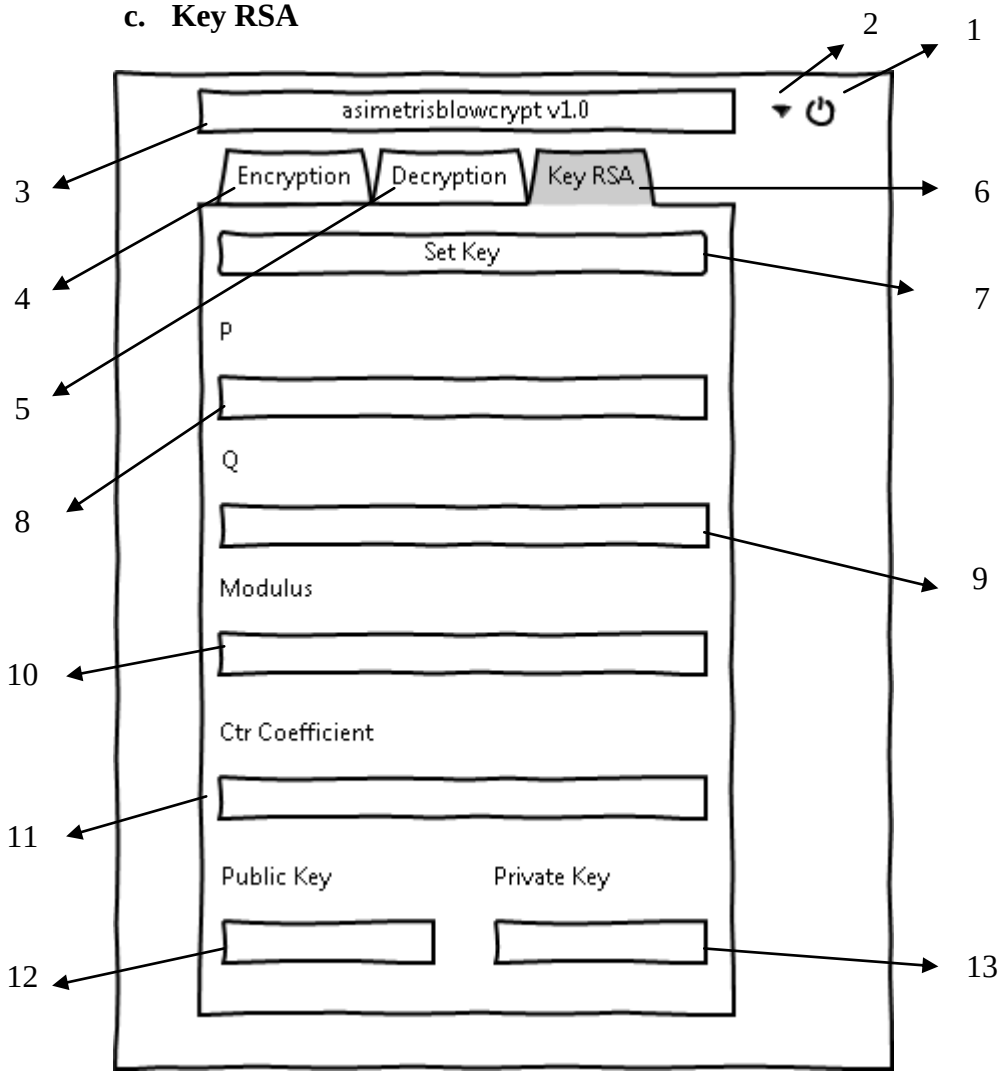
Gambar 4.8 Tampilan Antarmuka Halaman Dekripsi

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.

c. Key RSA



Gambar 4.9 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.

4.4 Perangkat Yang Digunakan

Perangkat yang digunakan dalam proses pembuatan aplikasi terdiri dari:

a. Perangkat Lunak
   1) Java (SE) Run Time Environment1.8.0_25.
   2) Netbeans IDE 7.4.
   3) Adobe Photoshop.
   4) Visual Paradigm For UML.
   5) Advance Installer 12.5

b. Perangkat Keras
   1) 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, dan Kapasitas Penyimpanan 320GB.
   2) Buku Teori dan Aplikasi Kriptografi.

BAB V

HASIL DAN PEMBAHASAN

5.1 HasilPengujianAplikasi

5.1.1. PengujianWhitebox

Pengujian yang bertujuan untuk meramalkan cara kerja perangkat lunak secara rinci, karenanya logicapath (jalur logika) perangkat lunak akan ditest dengan menyediakan testcase yang akan mengerjakan kondisi atau pengulangan secara spesifik. Secara sekilas dapat diambil kesimpulan whitebox testing merupakan petunjuk untuk mendapatkan program yang benar. Pengujian whitebox berfokus pada struktur kontrol program. Testcase dilakukan untuk memastikan bahwa semua statement pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji.

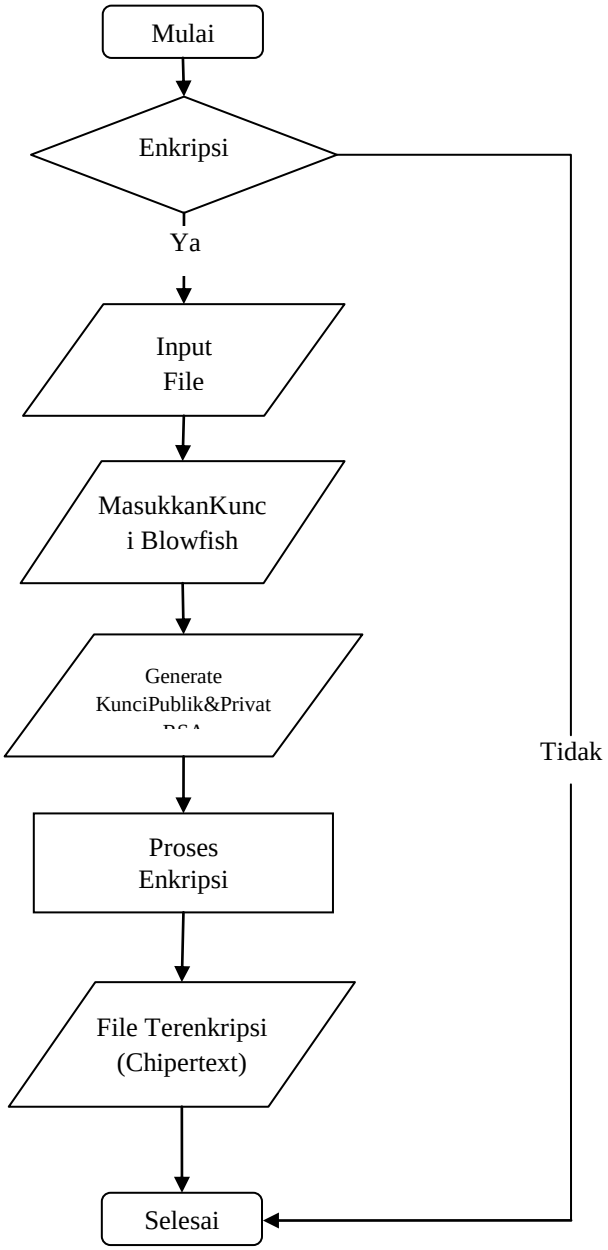
Sebagai contoh, akan dibahas pengujian terhadap enkripsi file. Secara garis besar, algoritmanya adalah sebagai berikut :

a. Melakukanenkripsifile.

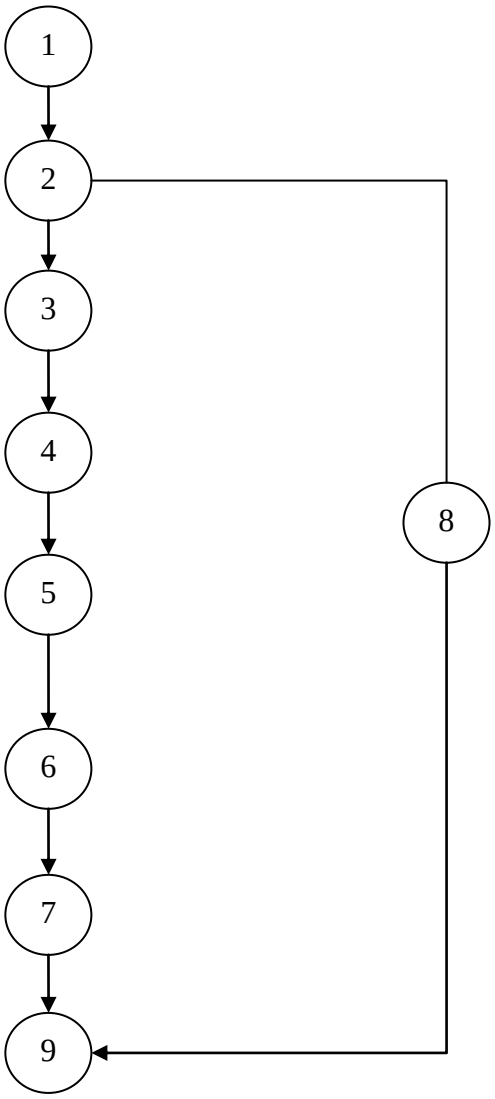
1. Penggunamemasukkan input berupafile. File yang akandiinputkanberupafileteks.
2. Masukkankunciblowfishuntukmengenkripsi.
3. Lakukanenkripsifileyang telahdiinputkan.
4. Fileyang telahterenkripsimenjadifile yang tidakterbaca (chipertext).
5. Chipertextdienkripsikembalidenganalgoritma RSA.

6. RSA kunci generator (kuncipublikdankunciprivat).
7. Enkripsiselesai..

Diagram alir enkripsi file adalah sebagai berikut :



Gambar 5.1 EnkripsiFile



Gambar 5.2 GrafikAlirEnkripsi File.

SkripPadaEnkripsi File

1. Skrippadaawalenkripsi file

```
/**
     * Creates new form Main
     */
    public Main() {
initComponents();
    }
```

1

2. Skripmemilih menu enkripsi

```
    jPanel1 = new javax.swing.JPanel();

// Code of sub-components - not shown here

// Layout setup code - not shown here

jTabbedPane1.addTab("Encryption", jPanel1);
```

2

3. Skrip input file

```
private void
btnbrowseActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:
JFileChooser chooser = new JFileChooser();
chooser.setCurrentDirectory(new java.io.File("."));
chooser.setFileFilter(new FileNameExtensionFilter("Plain Text
(.txt)", "txt"));
chooser.setDialogTitle("Choose File");
chooser.setFileSelectionMode(JFileChooser.FILES_AND_DIREC
TORIES);
chooser.setAcceptAllFileFilterUsed(false);

        if (chooser.showOpenDialog(null) ==
JFileChooser.APPROVE_OPTION) {
//      System.out.println("getCurrentDirectory(): " +
chooser.getCurrentDirectory());
//      System.out.println("getSelectedFile() : " +
chooser.getSelectedFile());
browsename.setText("" + chooser.getSelectedFile());
namafile = chooser.getSelectedFile().getName();
```

3

```
        try (BufferedReaderbr = new BufferedReader(new
FileReader("" + chooser.getSelectedFile()))) {

            while ((sCurrentLine = br.readLine()) != null) {
System.out.println(sCurrentLine);
                en = sCurrentLine;
                 dd1 = ""+chooser.getCurrentDirectory();
                 File as = chooser.getSelectedFile();

                 double h = as.length();
                 double j = h / 1024;
enfile.setText(""+j+"kb");
entujuan.setText(""+j+"kb");
            }

        } catch (IOException e) {
e.printStackTrace();
        } catch (Exception ex) {
Logger.getLogger(Main.class.getName()).log(Level.SEVERE,
null, ex);
        }
    } else {
System.out.println("No Selection ");
    }
  }
```

4. Skripmasukkankunci blowfish

```
private void passActionPerformed(java.awt.event.ActionEventevt)
{
    // TODO add your handling code here:
  }
```

4

5. Skrip generate kuncipublikdanprivat RSA

```
private void
setkeyActionPerformed(java.awt.event.ActionEventevt) {
      // TODO add your handling code here:
      try {
System.out.println("-------GENRATE PUBLIC and PRIVATE
KEY----------");
KeyPairGeneratorkeyPairGenerator =
KeyPairGenerator.getInstance("RSA");
keyPairGenerator.initialize(2048); //1024 used for normal securities
KeyPairkeyPair = keyPairGenerator.generateKeyPair();
PublicKeypublicKey = keyPair.getPublic();
PrivateKeyprivateKey = keyPair.getPrivate();
System.out.println("Public Key - " + publicKey);
System.out.println("Private Key - " + privateKey);

KeyFactorykeyFac = KeyFactory.getInstance("RSA");

RSAPrivateCrtKeySpecpkSpec = keyFac.getKeySpec(privateKey,
RSAPrivateCrtKeySpec.class);
System.out.println("Prime exponent p : " + pkSpec.getPrimeP());
System.out.println("Prime exponent q : " + pkSpec.getPrimeQ());
System.out.println("Modulus : " + pkSpec.getModulus());
System.out.println("Private exponent : " +
pkSpec.getPrivateExponent());
System.out.println("Public exponent : " +
pkSpec.getPublicExponent());

  //Pullingout parameters which makes up Key
System.out.println("\n------- PULLING OUT PARAMETERS
WHICH MAKES KEYPAIR----------\n");
KeyFactorykeyFactory = KeyFactory.getInstance("RSA");
RSAPublicKeySpecrsaPubKeySpec =
keyFactory.getKeySpec(publicKey, RSAPublicKeySpec.class);
RSAPrivateKeySpecrsaPrivKeySpec =
keyFactory.getKeySpec(privateKey, RSAPrivateKeySpec.class);
System.out.println("PubKeyModulus : " +
rsaPubKeySpec.getModulus());
System.out.println("PubKeyExponent : " +
rsaPubKeySpec.getPublicExponent());
System.out.println("PrivKeyModulus : " +
rsaPrivKeySpec.getModulus());
```

```
System.out.println("PrivKeyExponent : " +
rsaPrivKeySpec.getPrivateExponent());

pubkey.setText(""+ rsaPubKeySpec.getModulus());
privkey.setText(""+rsaPrivKeySpec.getPrivateExponent());
moduluskey.setText(""+rsaPubKeySpec.getModulus());
pkey.setText(""+pkSpec.getPrimeP());
qkey.setText(""+ pkSpec.getPrimeQ());


itungkey.setText(""+pkSpec.getCrtCoefficient());


  } catch (NoSuchAlgorithmException e) {
e.printStackTrace();
}catch (InvalidKeySpecException e) {
e.printStackTrace();
  }

   }
```

6. Skrip proses enkripsi

```
private void
encryptActionPerformed(java.awt.event.ActionEventevt) {
```



```
      try {
         // TODO add your handling code here:
         pas = pass.getText().toString();
         String pas2 = pass2.getText().toString();

         if (browsename.getText().toString().equals("")) {
JOptionPane.showMessageDialog(null, "File tidakbolehkosong!",
"Error",
JOptionPane.ERROR_MESSAGE);
         }
         if (pas.equals("")) {
JOptionPane.showMessageDialog(null, "Password harus di isi!",
"Error",
JOptionPane.ERROR_MESSAGE);
         }
         if (!pas.equals(pas2)) {
```

```
JOptionPane.showMessageDialog(null, "Password Tidaksama",
"Error",
JOptionPane.ERROR_MESSAGE);
          }

          if (!pas.equals("") &&pas.equals(pas2) &&
!browsename.getText().toString().equals("")) {
ActionListener al = new ActionListener() {
          public void actionPerformed(ActionEvent ae)
{setDuration();}
       };
       new Timer(10, al).start();
              for (inti = 0; i<= 100; i+=10) {
          final intcurrentValue = i;
SwingUtilities.invokeLater(new Runnable() {
                 public void run() {
pbs.setValue(currentValue);
                 }
              });
java.lang.Thread.sleep(100);
                 }
this.encrypt(pas, en);
System.out.println("" + en);
//               this.encryptData(en);
this.deencrypt("123", pas);
writing();
JOptionPane.showMessageDialog(null, "Encrypt Sukses\nCek File
Enkripsi di "+dd1+" andadengannama " + name2, "Sukses",
JOptionPane.INFORMATION_MESSAGE);
          }
       } catch (Exception ex) {
       }
   }
```

7. Skripnotifikasienkripsiselesai.

```
private void
encryptActionPerformed(java.awt.event.ActionEventevt) {
```

7

```
        try {
            // TODO add your handling code here:
            pas = pass.getText().toString();
            String pas2 = pass2.getText().toString();

            if (browsename.getText().toString().equals("")) {
JOptionPane.showMessageDialog(null, "File tidakbolehkosong!",
"Error",
JOptionPane.ERROR_MESSAGE);
            }
            if (pas.equals("")) {
JOptionPane.showMessageDialog(null, "Password harus di isi!",
"Error",
JOptionPane.ERROR_MESSAGE);
            }
            if (!pas.equals(pas2)) {
JOptionPane.showMessageDialog(null, "Password Tidaksama",
"Error",
JOptionPane.ERROR_MESSAGE);
            }

            if (!pas.equals("") &&pas.equals(pas2) &&
!browsename.getText().toString().equals("")) {
ActionListener al = new ActionListener() {
            public void actionPerformed(ActionEvent ae)
{setDuration();}
        };
        new Timer(10, al).start();
                for (inti = 0; i<= 100; i+=10) {
            final intcurrentValue = i;
SwingUtilities.invokeLater(new Runnable() {
                  public void run() {
pbs.setValue(currentValue);
                  }
               });
java.lang.Thread.sleep(100);
                }
```

```
this.encrypt(pas, en);
System.out.println("" + en);
//              this.encryptData(en);
this.deencrypt("123", pas);
writing();
JOptionPane.showMessageDialog(null, "Encrypt Sukses\nCek File
Enkripsi di "+dd1+" andadengannama " + name2, "Sukses",
JOptionPane.INFORMATION_MESSAGE);
          }
       } catch (Exception ex) {
       }
   }
```

8. Skripkeluaraplikasi

```
function closeWin() {
   myWindow.close();   //
}
```

8

Kompleksitas Siklomatis (pengukuran kuantitatif terhadap kompleksitas logis suatu program) dari grafik alir dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2$$

Dimana :

E = Jimlah edge grafik alir yang ditandakan dengan gambar panah

N = Jumlah simpul grafik alir yang ditandakan dengan gambar lingkaran

Sehingga kompleksitas siklomatisnya

V(G) = 8– 8+ 2 = 2

Basis set yang dihasilkan dari jalur independent secara linier adalah sebagai berikut :

1 – 2 – 3 – 4 – 5 – 6 – 7 – 9

1 – 2 – 8 – 9

Berdasarkan hasil tersebut, dari segi kelayakan software, sistem ini telah memenuhi syarat.

5.1.2. PengujianBlackbox

Metode yang digunakandalampengujianiniadalahpengujianblack box yang berfokuspadapersyaratanfungsionaldarisistem yang dibangun.

5.1.1.1. KasusdanHasilPengujian

1. Pengujian Menu Enkripsi

KelasUji	SkenarioUji	Hasil yang diharapkan	Kesimpulan
Menu Enkripsi	Memilih Tab Menu Enkripsi	Menampilkan Tab Menu Enkripsi	[√] Berhasil [ ] Tidakberhasil
	Memilih file yang akandienkripsi	Dapatmemilih file yang akandienkripsi	
	Memasukkan password	Dapatmemasukkan password	
	Generate Password	Dapatmelakukangenerate password	
	Menampilkaninformasiukuranfile	Ukuranfileditampilkan	
	Memilihtombolenkripsi	Memulai proses enkripsifile.	

Tabel 5.1 Pengujian Menu Enkripsi

2. Pengujian Menu Dekripsi

KelasUji	SkenarioUji	Hasil yang diharapkan	Kesimpulan
Menu Dekripsi	Memilih Tab Menu Dekripsi	Menampilkan Tab Menu Dekripsi	[√] Berhasil [ ] Tidakberhasil
	Memilih file yang akandidekripsi	Dapatmemilih file yang akandidekripsi	
	Memasukkan password	Dapatmemasukkan password	
	Menampilkaninformasiukuranfile	Ukuranfileditampilkan	
	Memilihtomboldekripsi	Memulai proses dekripsifile.	

Tabel 5.2Pengujian Menu Dekripsi

3. Pengujian Menu Key RSA

KelasUji	SkenarioUji	Hasil yang diharapkan	Kesimpulan
Menu Key RSA	Memilih Tab Menu Key RSA	Menampilkan Tab Menu Key RSA	[√] Berhasil [ ] Tidakberhasil
	MemilihtombolSet Key	DapatmemilihtombolSet Key	
	Generate nilai P	Dapatgeneratenilai P	
	Generate nilai Q	Dapatgeneratenilai Q	
	Generate nilai Modulus	Dapatgenerate nilai Modulus	
	Generate nilaiCtr Coefficient	Dapatgenerate nilaiCtr Coefficient	
	Generate nilai Public Key	Dapatgenerate nilai Public Key	
	Generate nilai Private Key	Dapatgenerate nilai Private Key	

Tabel 5.3Pengujian Menu Enkripsi

5.1.3. HasilPengujianEnkripsi File

Hasilpengujianenkripsi file denganaplikasiasimetrisblowfish v.1 adalahsebagaiberikut:

Plaintext	Ukuran FileOrginal (byte)	Ukuran File Enkripsi (byte)	Waktu (detik)
File1.txt	0.9765625	352	1
File2.txt	4320	344	1
File3.txt	123000	344	1

5.1.4. HasilPengujianDekripsi File

Hasilpengujianenkripsi file denganaplikasiasimetrisblowfish v.1 adalahsebagaiberikut:

Chipertext	Ukuran File Enkripsi (byte)	Ukuran FileOrginal (byte)	Waktu (detik)
File1.txt	352	1	1
File2.txt	344	4320	1
File3.txt	344	123000	1

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari analisaalgoritmadanimplementasi program makahasil yang didapatkandenganmenggunakanalgoritma Blowfish dan RSA dapatdisimpulkansebagaiberikut:

1. Enkripsidenganalgoritmablowfishterdiridariduabagian, yaituekspansikunci yang berfungsimerubahkunci (minimum 32-bit, maksimum 448-bit) menjadibeberapa array subkunci (subkey) dengan total 4168 byte (18x32-bit untukP-arraydan 4x256x32-bit untuk S-box sehinggatotalnya 33344 bit atay 4168 byte) danenkripsi data.
2. EnkripsidenganalgoritmaRSAmemilikiduabuahkunci yang berbeda, yaitukuncipublikdankunciprivat. PrinsipkerjaalgoritmaRSAmenggunakanoperasipemangkatandanoperasi mod (modulus) yang menghasilkannilai yang relatifacak.
3. UkuranfilesebelumdansesudahdienkripsidenganalgoritmaBlowfishdanRSAterjadiperubahankarenaadanyapenambahanukuranfilesesuaidenganalgoritma yang digunakan.

6.2. Saran

1. Disarankandalammendesainsistemkeamananhendaknyamengikutitaha-pan-tahapandasar yang benardari input proses dan output.
2. Dalammendesainsistemkeamananhendaknyadilakukanpemilihanalgoritma yang sesuaidanaman, sertawaktu yang efisien.
3. Disarankan agar bisamelakukanenkripsidandekripsiuntuktipe file lain.

DAFTAR PUSTAKA

Ariyus, Dony. 2006. **Kriptografi : Keamanan Data dan Komunikasi.** Yogyakarta : Graha Ilmu

Munir, Rinaldi. 2006. **Kriptografi**. Bandung : Informatika

Hakim S, Rachmad. 2010. **Buku Pintar Windows 7**. Jakarta: PT Elex Media Komputindo.

Pressman, Roger. 2005**. Software Engineering: A Practitioner's Approach**, Edisi ke 6. New York : McGraw-Hill.

Kadir, Abdul. 2012. **Algoritma & Pemrograman Menggunakan Java**. Jakarta: Andi.

Schildt, Herbert. 2005. **Java™: A Beginner's Guide, Third Edition.** New York: McGraw-Hill.

Swastika, Windra. Paulus Lucky. 2012. **Dasar Algoritma dan Pemrograman Menggunakan C dan Java**. Tanggerang: Prestasi Pustaka Publisher.

Wikipedia,**Blowfish(cipher)**.http://en.wikipedia.org/wiki/Blowfish_%28cipher%29 (diakses tanggal 1 April 2015)

Sitinjak, Suriski., Yuli Fauziah.,& Juwairiah. **Aplikasi Kriptografi File Menggunakan Algoritma Blowfish.** http://repository.upnyk.ac.id/395/1/C-12_APLIKASI_KRIPTOGRAFI_FILE_MENGGUNAKAN_ALGORITMA_BLOWFISH.pdf (diakses tanggal 1 April 2015)

Syafari, Anjar, 2007, Sekilas **Tentang Enkripsi Blowfish**, http://ilmukomputer.org/2007/07/27/sekilas-tentang-enkripsi-blowfish/ (diakses tanggal 1 April 2015)

Schneier, Bruce, 1996, **Applied Cryptography**, Second Edition, John Wiley & Son, New York.

Tambunan, Shanty Erikawaty Aryani. **Implementasi Algoritma Kriptografi Blowfish Untuk Keamanan Dokumen Pada Microsoft Office.**. http://repository.amikom.ac.id/index.php/detail/2213/IMPLEMENTASI%20ALGORITMA%20KRIPTOGRAFI%20BLOWFISH%20UNTUK%20KEAMANAN%20DOKUMEN%20PADA%20MICROSOFT%20OFFICE (diakses tanggal 1 April 2015)

Source Code

Main.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package encypt;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.math.BigInteger;
import java.security.KeyFactory;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.RSAPrivateCrtKeySpec;
import java.security.spec.RSAPrivateKeySpec;
import java.security.spec.RSAPublicKeySpec;
import java.util.Random;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.SwingUtilities;
import javax.swing.Timer;
import javax.swing.filechooser.FileNameExtensionFilter;
import org.apache.commons.codec.DecoderException;
```

```
import org.apache.commons.codec.binary.Hex;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

/**
 *
 * @author Noval Eka Herdinata
 */
public class Main extends javax.swing.JFrame {

    private static final String PUBLIC_KEY_FILE = "Public.key";
    private static final String PRIVATE_KEY_FILE = "Private.key";
    String sCurrentLine;
    String sCurrentLine2;
    String en, de;
    String namafile, namafile2;
    String pas, pasde, henkrip, senkrip;
byte[] encryptedData = null;
char[] encryptedTranspherable;
String[] aagh, aagi;
    String hu;
byte[] hasil;
    String d1,dd1, d2, name, name2;

    /**
     * Creates new form Main
     */
    public Main() {
initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jTabbedPane1 = new javax.swing.JTabbedPane();
        jPanel1 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        browsename = new javax.swing.JTextField();
```

```
btnbrowse = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
pass = new javax.swing.JPasswordField();
pass2 = new javax.swing.JPasswordField();
encrypt = new javax.swing.JButton();
jLabel11 = new javax.swing.JLabel();
gpass = new javax.swing.JButton();
pbs = new javax.swing.JProgressBar();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
enfile = new javax.swing.JLabel();
entujuan = new javax.swing.JLabel();
enwktu = new javax.swing.JLabel();
time = new javax.swing.JLabel();
jPanel2 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
browsenamede = new javax.swing.JTextField();
btnbrowsede = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
passde = new javax.swing.JPasswordField();
pass2de = new javax.swing.JPasswordField();
jLabel9 = new javax.swing.JLabel();
decrypt = new javax.swing.JButton();
jLabel10 = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
jLabel23 = new javax.swing.JLabel();
pbs1 = new javax.swing.JProgressBar();
ufile = new javax.swing.JLabel();
utujuan = new javax.swing.JLabel();
jPanel3 = new javax.swing.JPanel();
setkey = new javax.swing.JButton();
pkey = new javax.swing.JTextField();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
qkey = new javax.swing.JTextField();
jLabel17 = new javax.swing.JLabel();
moduluskey = new javax.swing.JTextField();
jLabel18 = new javax.swing.JLabel();
itungkey = new javax.swing.JTextField();
pubkey = new javax.swing.JTextField();
privkey = new javax.swing.JTextField();
jLabel19 = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jLabel1.setText("asimetrisblowcrypt v1.0");

        jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel3.setText("Browse File");

    browsename.setEnabled(false);

    btnbrowse.setText("Browse");
    btnbrowse.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
               btnbrowseActionPerformed(evt);
            }
        });

        jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel2.setText("Password");

        jLabel4.setText("Input Password: ");

        jLabel5.setText("Repeat Password:");

    pass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
               passActionPerformed(evt);
            }
        });

    encrypt.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    encrypt.setText("Encryption");
    encrypt.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
               encryptActionPerformed(evt);
            }
        });

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jLabel11.setText("Encryption");

    gpass.setText("Generate Password");
    gpass.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
               gpassActionPerformed(evt);
            }
```

```
        });

        jLabel12.setText("ukuran file sumber:");

        jLabel13.setText("ukuran file tujuan: ");

        jLabel14.setText("waktu proses");

    time.setText("00:00:00");

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
    .addGroup(jPanel1Layout.createSequentialGroup()
    .addContainerGap()
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
    .addComponent(jLabel3)
    .addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(browsename,
javax.swing.GroupLayout.PREFERRED_SIZE,                    260,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
    .addComponent(btnbrowse))
    .addComponent(jLabel2)
    .addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(jLabel4)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
    .addComponent(pass))
    .addGroup(jPanel1Layout.createSequentialGroup()
    .addComponent(jLabel5)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
    .addComponent(pass2))
    .addComponent(encrypt,      javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel11)
    .addComponent(gpass,        javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
        .addComponent(pbs, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel12)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(enfile))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING, false)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel14)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(time)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(enwktu))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel13)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(entujuan))))
        .addContainerGap(35, Short.MAX_VALUE))
            );
            jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel11)
        .addGap(18, 18, 18)
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(browsename,
javax.swing.GroupLayout.PREFERRED_SIZE,                23,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnbrowse))
        .addGap(18, 18, 18)
        .addComponent(jLabel2)
        .addGap(9, 9, 9)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
```

```
        .addComponent(jLabel4)
        .addComponent(pass,        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel5)
        .addComponent(pass2,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(gpass)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel12)
        .addComponent(enfile))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel13)
        .addComponent(entujuan))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel14)
        .addComponent(enwktu)
        .addComponent(time))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(pbs,    javax.swing.GroupLayout.DEFAULT_SIZE,    31,
Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(encrypt,    javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18))
            );

            jTabbedPane1.addTab("Encryption", jPanel1);

            jLabel6.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
            jLabel6.setText("Browse File");

        browsenamede.setEnabled(false);

        btnbrowsede.setText("Browse");
        btnbrowsede.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnbrowsedeActionPerformed(evt);
            }
        });

            jLabel7.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
            jLabel7.setText("Password");

            jLabel8.setText("Input Password: ");

            jLabel9.setText("Repeat Password:");

        decrypt.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        decrypt.setText("Decryption");
        decrypt.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                decryptActionPerformed(evt);
            }
        });

            jLabel10.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
            jLabel10.setText("Decryption");

            jLabel21.setText("ukuran file sumber:");

            jLabel22.setText("ukuran file tujuan: ");

            jLabel23.setText("waktu proses");

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
            jPanel2.setLayout(jPanel2Layout);
            jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
```

```
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(jLabel21)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(ufile))
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addComponent(jLabel22)
        .addComponent(jLabel23))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(utujuan)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING)
        .addComponent(pbs1,     javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup()
        .addComponent(jLabel8)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(passde))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup()
        .addComponent(jLabel9)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(pass2de))
        .addComponent(decrypt,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING)
        .addComponent(jLabel6,
javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup()
```

```
        .addComponent(browsenamede,
javax.swing.GroupLayout.PREFERRED_SIZE,                    260,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(btnbrowsede))
        .addComponent(jLabel7,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel10,
javax.swing.GroupLayout.Alignment.LEADING))
        .addGap(0, 0, Short.MAX_VALUE)))
        .addGap(35, 35, 35))))
              );
              jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel10)
        .addGap(18, 18, 18)
        .addComponent(jLabel6)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(browsenamede,
javax.swing.GroupLayout.PREFERRED_SIZE,                    23,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnbrowsede))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(jLabel7)
        .addGap(9, 9, 9)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel8)
        .addComponent(passde,     javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel9)
```

```
        .addComponent(pass2de,   javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel21)
        .addComponent(ufile))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel22)
        .addComponent(utujuan))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(jLabel23)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(pbs1,    javax.swing.GroupLayout.DEFAULT_SIZE,    31,
Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(decrypt,    javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(54, 54, 54))
              );

              jTabbedPane1.addTab("Decryption", jPanel2);

        setkey.setText("Set Key");
        setkey.addActionListener(new java.awt.event.ActionListener() {
              public void actionPerformed(java.awt.event.ActionEvent evt) {
                 setkeyActionPerformed(evt);
              }
        });

        jLabel15.setText("P");

        jLabel16.setText("Q");

        jLabel17.setText("Modulus");

        moduluskey.addActionListener(new java.awt.event.ActionListener() {
              public void actionPerformed(java.awt.event.ActionEvent evt) {
                 moduluskeyActionPerformed(evt);
              }
```

```
        });

        jLabel18.setText("Ctr Coefficient");

    itungkey.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
              itungkeyActionPerformed(evt);
            }
        });

    pubkey.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
              pubkeyActionPerformed(evt);
            }
        });

    privkey.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
              privkeyActionPerformed(evt);
            }
        });

        jLabel19.setText("Public Key");

        jLabel20.setText("Private Key");

    javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
    .addGroup(jPanel3Layout.createSequentialGroup()
    .addContainerGap()
    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
    .addComponent(setkey, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(pkey)
    .addComponent(qkey)
    .addComponent(moduluskey)
    .addComponent(itungkey)
    .addGroup(jPanel3Layout.createSequentialGroup()
    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
    .addComponent(pubkey, javax.swing.GroupLayout.PREFERRED_SIZE,
176, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel15)
        .addComponent(jLabel16)
        .addComponent(jLabel17)
        .addComponent(jLabel18)
        .addComponent(jLabel19))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
        .addComponent(jLabel20)
        .addGap(0, 121, Short.MAX_VALUE))
        .addComponent(privkey))))
        .addContainerGap())
            );
            jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
        .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(setkey)
        .addGap(7, 7, 7)
        .addComponent(jLabel15)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(pkey,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(jLabel16)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(qkey,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(jLabel17)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(moduluskey,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
```

```
        .addComponent(jLabel18)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(itungkey,  javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(7, 7, 7)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel19)
        .addComponent(jLabel20))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(pubkey,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(privkey,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(126, Short.MAX_VALUE))
            );

            jTabbedPane1.addTab("Key RSA", jPanel3);

        javax.swing.GroupLayout          layout          =          new
javax.swing.GroupLayout(getContentPane());
            getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addComponent(jTabbedPane1)
        .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );
        layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(jTabbedPane1))
```

```
            );

        pack();
           }// </editor-fold>//GEN-END:initComponents

           @SuppressWarnings("empty-statement")
           private void btnbrowsedeActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_btnbrowsedeActionPerformed
               // TODO add your handling code here:
               JFileChooser chooser = new JFileChooser();
        chooser.setCurrentDirectory(new java.io.File("."));
        chooser.setFileFilter(new FileNameExtensionFilter("Plain Text (.txt)",
"txt"));
        chooser.setDialogTitle("Choose File");
        chooser.setFileSelectionMode(JFileChooser.FILES_AND_DIRECTORIE
S);
        chooser.setAcceptAllFileFilterUsed(false);

               if (chooser.showOpenDialog(null) ==
JFileChooser.APPROVE_OPTION) {
        //                  System.out.println("getCurrentDirectory(): " +
chooser.getCurrentDirectory());
        //     System.out.println("getSelectedFile() : " + chooser.getSelectedFile());
        browsenamede.setText("" + chooser.getSelectedFile());
                    namafile2 = chooser.getSelectedFile().getName();

                    File oldfile =new File(""+chooser.getSelectedFile());
                        File newfile =new
File(""+chooser.getSelectedFile()+".txt");

                        if(oldfile.renameTo(newfile)){
                            try (BufferedReader br = new BufferedReader(new
FileReader("" + chooser.getSelectedFile()+".txt"))) {

                    while ((sCurrentLine2 = br.readLine()) != null) {
                       System.out.println(sCurrentLine2);


                        File ads = chooser.getSelectedFile();

                        double h = ads.length();
                        double j = h / 1024;
        ufile.setText(""+j+"kb");
        utujuan.setText(""+j+"kb");


                        String as = sCurrentLine2;
                                String output = "";
```

```
        for(int i = 0; i <= as.toString().length() - 8; i+=8)
        {
           int k = Integer.parseInt(as.toString().substring(i, i+8), 2);
           output += (char) k;
        }

                        de = output;
                        aagh = de.split(",", 6);
                        System.out.println("" + aagh[2]);
                        d1 = ""+chooser.getCurrentDirectory();
                        d2 = ""+chooser.getSelectedFile();
                        System.out.println(""+d1);


        decrypto("" + aagh[0]);
                    }

                } catch (IOException e) {
        e.printStackTrace();
                } catch (Exception ex) {
                   Logger.getLogger(Main.class.getName()).log(Level.SEVERE,
null, ex);
                }
                        }else{
                                System.out.println("Rename failed");
                        }



            } else {
               System.out.println("No Selection ");
            }
          }//GEN-LAST:event_btnbrowsedeActionPerformed

           private void btnbrowseActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_btnbrowseActionPerformed
               // TODO add your handling code here:
               JFileChooser chooser = new JFileChooser();
        chooser.setCurrentDirectory(new java.io.File("."));
        chooser.setFileFilter(new FileNameExtensionFilter("Plain Text (.txt)",
"txt"));
        chooser.setDialogTitle("Choose File");
        chooser.setFileSelectionMode(JFileChooser.FILES_AND_DIRECTORIE
S);
        chooser.setAcceptAllFileFilterUsed(false);

               if (chooser.showOpenDialog(null) ==
JFileChooser.APPROVE_OPTION) {
```

```
        //                      System.out.println("getCurrentDirectory():  "  +
chooser.getCurrentDirectory());
        //      System.out.println("getSelectedFile() : " + chooser.getSelectedFile());
        browsename.setText("" + chooser.getSelectedFile());
                namafile = chooser.getSelectedFile().getName();


                try (BufferedReader br = new BufferedReader(new FileReader(""
+ chooser.getSelectedFile()))) {

                    while ((sCurrentLine = br.readLine()) != null) {
                       System.out.println(sCurrentLine);
                       en = sCurrentLine;
                        dd1 = ""+chooser.getCurrentDirectory();
                        File as = chooser.getSelectedFile();

                        double h = as.length();
                        double j = h / 1024;
        enfile.setText(""+j+"kb");
        entujuan.setText(""+j+"kb");
                    }

                } catch (IOException e) {
        e.printStackTrace();
                } catch (Exception ex) {
                    Logger.getLogger(Main.class.getName()).log(Level.SEVERE,
null, ex);
                }
            } else {
                System.out.println("No Selection ");
            }
          }//GEN-LAST:event_btnbrowseActionPerformed

           public void setDuration(){
             //menghitung selisih waktu start dengan waktu sekarang
               int s = 0, m = 0, h = 0;
             if(s==60){
                s = 0;
                m++;
        }else s++;
              if(m==60){
                m = 0;
                h++;
             }
             Time t = new Time();
             TimeEntity te = t.timeFormat(s, m, h);
        time.setText(te.getJam()+" : "+te.getMenit()+" : "+te.getDetik());
          }
```

```
        private void encryptActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_encryptActionPerformed


            try {
                // TODO add your handling code here:
                pas = pass.getText().toString();
                String pas2 = pass2.getText().toString();

                if (browsename.getText().toString().equals("")) {
                    JOptionPane.showMessageDialog(null, "File tidak boleh kosong!", "Error",
                            JOptionPane.ERROR_MESSAGE);
                }
                if (pas.equals("")) {
                    JOptionPane.showMessageDialog(null, "Password harus di isi!", "Error",
                            JOptionPane.ERROR_MESSAGE);
                }
                if (!pas.equals(pas2)) {
                    JOptionPane.showMessageDialog(null, "Password Tidak sama", "Error",
                            JOptionPane.ERROR_MESSAGE);
                }

                 if (pass.getText().length() <=8 ) {
                    JOptionPane.showMessageDialog(null, "Password minimal 8 karakter", "Error",
                            JOptionPane.ERROR_MESSAGE);
                }

                if (!pas.equals("") && pas.equals(pas2) && !browsename.getText().toString().equals("") && pass.getText().length() >=8) {
                     ActionListener al = new ActionListener() {
                    public void actionPerformed(ActionEvent ae) {setDuration();}
                };
                new Timer(10, al).start();
                      for (int i = 0; i <= 100; i+=10) {
                    final int currentValue = i;
                    SwingUtilities.invokeLater(new Runnable() {
                        public void run() {
          pbs.setValue(currentValue);
                        }
                    });
          java.lang.Thread.sleep(100);
                    }
```

```
        this.encrypt(pas, en);
                System.out.println("" + en);
        //          this.encryptData(en);
        this.deencrypt("123", pas);
        writing();
                JOptionPane.showMessageDialog(null, "Encrypt Sukses\nCek
File Enkripsi di "+dd1+" anda dengan nama " + name2, "Sukses",
                    JOptionPane.INFORMATION_MESSAGE);
            }
        } catch (Exception ex) {
        }
    }//GEN-LAST:event_encryptActionPerformed

    private void gpassActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_gpassActionPerformed
        // TODO add your handling code here:

        String x = null;


         Random r = new Random();

    String alphabet = "123456789abcdefghijklmnopqrstuvwxyz!^&*()";
    for (int i = 0; i < 50; i++) {
      x                                                                  =
""+alphabet.charAt(r.nextInt(alphabet.length()))+""+alphabet.charAt(r.nextInt(alp
habet.length()))+""+alphabet.charAt(r.nextInt(alphabet.length()))+""+alphabet.cha
rAt(r.nextInt(alphabet.length()))+""+alphabet.charAt(r.nextInt(alphabet.length()))
+""+alphabet.charAt(r.nextInt(alphabet.length()))+""+alphabet.charAt(r.nextInt(al
phabet.length()))+""+alphabet.charAt(r.nextInt(alphabet.length()))+""+alphabet.c
harAt(r.nextInt(alphabet.length()));
    } // prints 50 random characters from alphabet

        pass.setText("" + x);
        pass2.setText("" + x);
        JOptionPane.showMessageDialog(null, "Pasword Anda: " + x,
"Sukses",
                JOptionPane.INFORMATION_MESSAGE);
    }//GEN-LAST:event_gpassActionPerformed

    private void decryptActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_decryptActionPerformed
        // TODO add your handling code here:

        pasde = passde.getText().toString();
        String pas2 = pass2de.getText().toString();

        if (browsenamede.getText().toString().equals("")) {
```

```
                JOptionPane.showMessageDialog(null, "File tidak boleh kosong!",
"Error",
                        JOptionPane.ERROR_MESSAGE);
                }
                if (pasde.equals("")) {
                JOptionPane.showMessageDialog(null, "Password harus di isi!",
"Error",
                        JOptionPane.ERROR_MESSAGE);
                }
                if (!pasde.equals(pas2)) {
                JOptionPane.showMessageDialog(null, "Password Tidak sama",
"Error",
                        JOptionPane.ERROR_MESSAGE);
                }

                  if (passde.getText().length() <=8 ) {
                     JOptionPane.showMessageDialog(null, "Password minimal 8
karakter", "Error",
                             JOptionPane.ERROR_MESSAGE);
                   }
                if                          (pasde.equals(pas2)                          &&
!browsenamede.getText().toString().equals("")&& passde.getText().length() >=8)
{
                   if (pasde.equals(aagh[1])) {

                        ActionListener al = new ActionListener() {
                   public void actionPerformed(ActionEvent ae) {setDuration();}
                };
                new Timer(10, al).start();
                     try {

                        for (int i = 0; i <= 100; i+=10) {
                   final int currentValue = i;
                   SwingUtilities.invokeLater(new Runnable() {
                        public void run() {
                           pbs1.setValue(currentValue);
                        }
                      });
          java.lang.Thread.sleep(100);
                       }
                        decrypt(aagh[2]);
                         JOptionPane.showMessageDialog(null,                "Decrypt
Sukses\nCek File Enkripsi di "+d1+" anda dengan nama " + aagh[4], "Sukses",
                             JOptionPane.INFORMATION_MESSAGE);
                      } catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
                      }
```

```
        }else{
                    JOptionPane.showMessageDialog(null, "Password Salah!",
"Error",
                      JOptionPane.ERROR_MESSAGE);
                  }
                }

            }//GEN-LAST:event_decryptActionPerformed

            private void moduluskeyActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_moduluskeyActionPerformed
                // TODO add your handling code here:
            }//GEN-LAST:event_moduluskeyActionPerformed

            private void itungkeyActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_itungkeyActionPerformed
                // TODO add your handling code here:
            }//GEN-LAST:event_itungkeyActionPerformed

            private void pubkeyActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_pubkeyActionPerformed
                // TODO add your handling code here:
            }//GEN-LAST:event_pubkeyActionPerformed

            private void privkeyActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_privkeyActionPerformed
                // TODO add your handling code here:
            }//GEN-LAST:event_privkeyActionPerformed

            private void setkeyActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_setkeyActionPerformed
                // TODO add your handling code here:
                try {
           System.out.println("-------GENRATE PUBLIC and PRIVATE KEY-----
--------");
           KeyPairGenerator                keyPairGenerator                =
KeyPairGenerator.getInstance("RSA");
           keyPairGenerator.initialize(2048); //1024 used for normal securities
           KeyPair keyPair = keyPairGenerator.generateKeyPair();
           PublicKey publicKey = keyPair.getPublic();
           PrivateKey privateKey = keyPair.getPrivate();
           System.out.println("Public Key - " + publicKey);
           System.out.println("Private Key - " + privateKey);

            KeyFactory keyFac = KeyFactory.getInstance("RSA");

           RSAPrivateCrtKeySpec pkSpec = keyFac.getKeySpec(privateKey,
RSAPrivateCrtKeySpec.class);
```

```
                System.out.println("Prime exponent p : " + pkSpec.getPrimeP());
                System.out.println("Prime exponent q : " + pkSpec.getPrimeQ());
                System.out.println("Modulus : " + pkSpec.getModulus());
                System.out.println("Private exponent : " + pkSpec.getPrivateExponent());
                System.out.println("Public exponent : " + pkSpec.getPublicExponent());

            //Pullingout parameters which makes up Key
            System.out.println("\n------- PULLING OUT PARAMETERS WHICH MAKES KEYPAIR----------\n");
            KeyFactory keyFactory = KeyFactory.getInstance("RSA");
            RSAPublicKeySpec rsaPubKeySpec = keyFactory.getKeySpec(publicKey, RSAPublicKeySpec.class);
            RSAPrivateKeySpec rsaPrivKeySpec = keyFactory.getKeySpec(privateKey, RSAPrivateKeySpec.class);
            System.out.println("PubKey Modulus : " + rsaPubKeySpec.getModulus());
            System.out.println("PubKey Exponent : " + rsaPubKeySpec.getPublicExponent());
            System.out.println("PrivKey Modulus : " + rsaPrivKeySpec.getModulus());
            System.out.println("PrivKey Exponent : " + rsaPrivKeySpec.getPrivateExponent());

        pubkey.setText(""+ rsaPubKeySpec.getModulus());
        privkey.setText(""+rsaPrivKeySpec.getPrivateExponent());
        moduluskey.setText(""+rsaPubKeySpec.getModulus());
        pkey.setText(""+pkSpec.getPrimeP());
        qkey.setText(""+ pkSpec.getPrimeQ());


        itungkey.setText(""+pkSpec.getCrtCoefficient());


          } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
        }catch (InvalidKeySpecException e) {
        e.printStackTrace();
          }

            }//GEN-LAST:event_setkeyActionPerformed

            private void passActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_passActionPerformed
                // TODO add your handling code here:
            }//GEN-LAST:event_passActionPerformed
```

```
    public void writing() {
        try {
            //Whatever the file path is.
            File statText = new File(""+dd1+"/"+namafile+" (Encrypte).txt");
            name2 = ""+namafile+" (Encrypte).txt";
            FileOutputStream is = new FileOutputStream(statText);
            OutputStreamWriter osw = new OutputStreamWriter(is);
            Writer w = new BufferedWriter(osw);
            String a = senkrip + "," + pas + "," + henkrip + "," +
en+","+namafile;

        byte[] bytes = a.getBytes();
          StringBuilder binary = new StringBuilder();
          for (byte b : bytes)
          {
            int val = b;
            for (int i = 0; i < 8; i++)
            {
        binary.append((val & 128) == 0 ? 0 : 1);
              val <<= 1;
            }
          }

            //Descypt Data using Private Key
        //          decryptData(encryptedData);
        w.write("" + binary);
        w.close();
            } catch (IOException e) {
                System.err.println("Problem writing to the file statsTest.txt");
            }
        }

        public void writing2() {
            try {
                //Whatever the file path is.
                System.out.println(""+d2);
                File statText = new File(""+d2+".txt");
                FileOutputStream is = new FileOutputStream(statText);
                OutputStreamWriter osw = new OutputStreamWriter(is);
                Writer w = new BufferedWriter(osw);
                String ad = ""+new String(hasil);
                System.out.println(""+ad);
        w.write("" + ad);
        w.close();
                    File oldfile =new File(""+d2+".txt");
                        File newfile =new File(""+d1+"/"+aagh[4]);
                        name = ""+d1+"/"+aagh[4];
                        if(oldfile.renameTo(newfile)){
```

```
                    System.out.println("sukses");
                }

            } catch (IOException e) {
                System.err.println("Problem writing to the file statsTest.txt");
            }
        }

        private void encrypt(String username, String password) throws
Exception {
        byte[] keyData = (username + password).getBytes();
            SecretKeySpec secretKeySpec = new SecretKeySpec(keyData,
"Blowfish");
            Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);
        byte[] hasil = cipher.doFinal(password.getBytes());
            System.out.println(new BASE64Encoder().encode(hasil));
            henkrip = "" + new BASE64Encoder().encode(hasil);
        }

        private void deencrypt(String username, String password) throws
Exception {
        byte[] keyData = (username + password).getBytes();
            SecretKeySpec secretKeySpec = new SecretKeySpec(keyData,
"Blowfish");
            Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);
        byte[] hasil = cipher.doFinal(password.getBytes());
            System.out.println(new BASE64Encoder().encode(hasil));
            senkrip = "" + new BASE64Encoder().encode(hasil);
        }

        private void decrypto(String string) throws Exception {
        byte[] keyData = ("123" + "" + aagh[1]).getBytes();
            SecretKeySpec secretKeySpec = new SecretKeySpec(keyData,
"Blowfish");
            Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.DECRYPT_MODE, secretKeySpec);
        byte[] hasil = cipher.doFinal(new
BASE64Decoder().decodeBuffer(string));
            System.out.println(new String(hasil));
        }

        private void decrypt(String string) throws Exception {
        byte[] keyData = (aagh[1] + aagh[3]).getBytes();
            SecretKeySpec secretKeySpec = new SecretKeySpec(keyData,
"Blowfish");
            Cipher cipher = Cipher.getInstance("Blowfish");
```

```
        cipher.init(Cipher.DECRYPT_MODE, secretKeySpec);
            hasil = cipher.doFinal(new BASE64Decoder().decodeBuffer(string));
            System.out.println(new String(hasil));
            writing2();
        }

        /**
         * @param args the command line arguments
         */
        public static void main(String args[]) {
            /* Set the Nimbus look and feel */
            //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
            /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
             * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
             */
            try {
                for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                    if ("Nimbus".equals(info.getName())) {
        javax.swing.UIManager.setLookAndFeel(info.getClassName());
                        break;
                    }
                }
            } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            }
            //</editor-fold>

            /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new Main().setVisible(true);
                }
            });
        }
```

```
    private byte[] encryptData(String data) throws IOException {
        System.out.println("\n----------------ENCRYPTION STARTED------------");

        System.out.println("Data Before Encryption :" + data);
    byte[] dataToEncrypt = data.getBytes("UTF-8");

        try {
            PublicKey pubKey = readPublicKeyFromFile(PUBLIC_KEY_FILE);
            Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.ENCRYPT_MODE, pubKey);
            encryptedData = cipher.doFinal(dataToEncrypt);
            System.out.println("Encryted Data: " + encryptedData);
        } catch (Exception e) {
    e.printStackTrace();
        }

        System.out.println("----------------ENCRYPTION COMPLETED------------");
        return encryptedData;
    }

    /**
     * Encrypt Data
     *
     * @param data
     * @throws IOException
     */
    private void decryptData(byte[] data) throws IOException {
        System.out.println("\n----------------DECRYPTION STARTED------------");
    byte[] descryptedData = null;

        try {
            PrivateKey privateKey = readPrivateKeyFromFile(PRIVATE_KEY_FILE);
            Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.DECRYPT_MODE, privateKey);
            descryptedData = cipher.doFinal(data);
            System.out.println("Decrypted Data: " + new String(descryptedData));

        } catch (Exception e) {
    e.printStackTrace();
        }
```

```
            System.out.println("----------------DECRYPTION COMPLETED------
------");
        }

        /**
         * read Public Key From File
         *
         * @param fileName
         * @return PublicKey
         * @throws IOException
         */
        public PublicKey readPublicKeyFromFile(String fileName) throws
IOException {
            FileInputStream fis = null;
            ObjectInputStream ois = null;
            try {
                fis = new FileInputStream(new File(fileName));
                ois = new ObjectInputStream(fis);

                BigInteger modulus = (BigInteger) ois.readObject();
                BigInteger exponent = (BigInteger) ois.readObject();

                //Get Public Key
                RSAPublicKeySpec rsaPublicKeySpec = new
RSAPublicKeySpec(modulus, exponent);
                KeyFactory fact = KeyFactory.getInstance("RSA");
                PublicKey publicKey = fact.generatePublic(rsaPublicKeySpec);

                return publicKey;

            } catch (Exception e) {
        e.printStackTrace();
            } finally {
                if (ois != null) {
        ois.close();
                    if (fis != null) {
        fis.close();
                    }
                }
            }
            return null;
        }

        /**
         * read Public Key From File
         *
         * @param fileName
         * @return
```

```
     * @throws IOException
     */
    public PrivateKey readPrivateKeyFromFile(String fileName) throws
IOException {
        FileInputStream fis = null;
        ObjectInputStream ois = null;
        try {
            fis = new FileInputStream(new File(fileName));
            ois = new ObjectInputStream(fis);

            BigInteger modulus = (BigInteger) ois.readObject();
            BigInteger exponent = (BigInteger) ois.readObject();

            //Get Private Key
            RSAPrivateKeySpec rsaPrivateKeySpec = new
RSAPrivateKeySpec(modulus, exponent);
            KeyFactory fact = KeyFactory.getInstance("RSA");
            PrivateKey privateKey = fact.generatePrivate(rsaPrivateKeySpec);

            return privateKey;

        } catch (Exception e) {
    e.printStackTrace();
        } finally {
            if (ois != null) {
    ois.close();
                if (fis != null) {
    fis.close();
                }
            }
        }
        return null;
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JTextField browsename;
    private javax.swing.JTextField browsenamede;
    private javax.swing.JButton btnbrowse;
    private javax.swing.JButton btnbrowsede;
    private javax.swing.JButton decrypt;
    private javax.swing.JButton encrypt;
    private javax.swing.JLabel enfile;
    private javax.swing.JLabel entujuan;
    private javax.swing.JLabel enwktu;
    private javax.swing.JButton gpass;
    private javax.swing.JTextField itungkey;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
```

```
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JTextField moduluskey;
    private javax.swing.JPasswordField pass;
    private javax.swing.JPasswordField pass2;
    private javax.swing.JPasswordField pass2de;
    private javax.swing.JPasswordField passde;
    private javax.swing.JProgressBar pbs;
    private javax.swing.JProgressBar pbs1;
    private javax.swing.JTextField pkey;
    private javax.swing.JTextField privkey;
    private javax.swing.JTextField pubkey;
    private javax.swing.JTextField qkey;
    private javax.swing.JButton setkey;
    private javax.swing.JLabel time;
    private javax.swing.JLabel ufile;
    private javax.swing.JLabel utujuan;
    // End of variables declaration//GEN-END:variables

}
```

TimeEntity.Java

```
/*
```

```
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package encypt;

/**
 *
 * @author
 */
public class TimeEntity {

    private String detik;
    private String menit;
    private String jam;

    public TimeEntity() {
    }

    public TimeEntity(String detik, String menit, String jam) {
this.detik = detik;
this.menit = menit;
        this.jam = jam;
    }

    public String getDetik() {
        return detik;
    }

    public void setDetik(String detik) {
this.detik = detik;
    }

    public String getJam() {
        return jam;
    }

    public void setJam(String jam) {
        this.jam = jam;
    }

    public String getMenit() {
        return menit;
    }

    public void setMenit(String menit) {
this.menit = menit;
    }
}
```

Time.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package encypt;

import java.util.Date;

/**
 *
 * @author
 */
public class Time {

    private Date dt;
    private String detik;
    private String menit;
    private String jam;

    public TimeEntity currTime(){
        String nol_jam = "";
        String nol_menit = "";
        String nol_detik = "";
        dt = new Date();
        TimeEntity tm = timeFormat(dt.getSeconds(), dt.getMinutes(),
dt.getHours());
        return tm;
    }

    public TimeEntity timeFormat(int s, int m, int h){
        TimeEntity te;
        String nolS="", nolM="", nolH="";
        if (s <= 9) nolS = "0";
        if (m <= 9) nolM = "0";
        if (h <= 9) nolH = "0";
        te = new TimeEntity(nolS+Integer.toString(s),
nolM+Integer.toString(m), nolH+Integer.toString(h));
        return te;
    }

}
```