PENERAPAN ALGORITMA RIVERT CODE 4 (RC 4) PADA APLIKASI KRIPTOGRAFI DOKUMEN

Harni Kusniyati[1], Satya Diansyah[2], Raka Yusuf[3]
Jurusan Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jl.Raya Meruya Selatan, Kembangan, Jakarta 11650
Email : harni.kusniyati@mercubuana.ac.id[1], satyadiansyah@gmail.com[2], raka@mercubauna.ac.id[3]

Abstract

The development of information technology, has made the information as a basic requirement for everyone. To secure the information we have, one of the techniques of data and information security is cryptography. Therefore, the authors make an application that can maintain the confidentiality of the information and the intended application is web-based cryptographic applications. This application can be used to secure the data. In this application, the cryptographic algorithm to be used is the algorithm Rivest Code 4 (RC4). RC4 is a stream cipher algorithm that processes the type of data input unit. Algorithms Rivest Code 4 (RC4) is also part of a symmetric algorithm, in which the encryption and decryption process has the same key. Making these applications using the programming language PHP and MySQL. Modeling methods in making this application is a method of UML (Unified Modeling Language). The results to be achieved from this research is biased document cryptographic applications perform encryption and decryption algorithms Rivest document with Code 4 (RC4).

Keywords: *Cryptography, Algorithms Rivest Code 4 (RC4), Encryption, Decryption*

Abstrak

Perkembangan teknologi informasi saat ini, telah menjadikan informasi sebagai kebutuhan pokok bagi setiap orang. Untuk mengamankan informasi yang kita punya, salah satu teknik pengamanan data dan informasi adalah kriptografi. Oleh karena itu penulis membuat suatu aplikasi yang dapat menjaga kerahasiaan informasi dan aplikasi yang dimaksud adalah aplikasi kriptografi berbasis web. Aplikasi ini dapat digunakan untuk mengamankan data. Pada aplikasi ini algoritma kriptografi yang akan digunakan adalah algoritma Rivest Code 4 (RC4). RC4 merupakan algoritma jenis stream cipher yang memproses unit input data. Algoritma Rivest Code 4 (RC4) juga merupakan bagian dari algoritma simetris, dimana proses enkripsi dan dekripsinya memiliki kunci yang sama. Pembuatan aplikasi ini menggunakan bahasa pemrograman PHP dan MySQL. Metode pemodelan dalam pembuatan aplikasi ini adalah metode UML (Unified Modelling Language). Hasil yang akan dicapai dari penelitian ini adalah aplikasi kriptografi dokumen yang bias melakukan enkripsi dan dekripsi dokumen dengan algoritma Rivest Code 4 (RC 4).

Kata kunci : *Kriptografi, Algoritma Rivest Code 4 (RC4), Enkripsi, Dekripsi.*

1. PENDAHULUAN

1.1. Latar Belakang

Pada era teknologi informasi sekarang ini, keamanan dalam penyimpanan dan pengiriman data atau informasi adalah hal yang sangat penting dan tidak dapat diabaikan. Terlebih jika pesan yang disimpan dan dikirim bersifat penting dan rahasia. Dengan makin berkembangnya teknologi yang begitu pesat maka bertukar informasi menjadi hal yang sangat mudah dengan hanya mengandalkan internet sebagai media pertukaran. Salah satu dampak negatif dalam perkembangan teknologi informasi adalah adanya pencurian data. Dengan adanya pencurian data maka aspek keamanan dalam pertukaran informasi serta penyimpanan data dianggap penting, karena suatu komunikasi data jarak jauh belum tentu aman dari pencurian.

Karena begitu pentingnya sebuah informasi, maka dibutuhkan suatu metode yang dapat menjaga kerahasiaan informasi dan metode yang dimaksud adalah kriptografi. Kriptgrafi bertujuan agar data atau informasi yang dikirim tidak dapat dibaca oleh orang yang tidak berhak. Dalam kriptografi ada yang disebut dengan enkripsi (encryption) yaitu proses penyamaran data dari plaintext (data asli) menjadi ciphertext (data tersandi) dan dekripsi (decryption) yaitu proses pengembalian ciphertext menjadi plaintext kembali. Kriptografi dipercaya untuk menangani masalah keamanan suatu data atau informasi dalam proses pengiriman, penyimpanan dan keperluan lainnya agar data atau informasi tetap terjaga kerahasiaannya.

Algoritma kriptografi ada dua tipe berdasarkan kuncinya, yaitu algoritma simetris yaitu algoritma yang menggunakan kunci yang sama untuk proses enkripsi serta dekripsi dan algoritma asimetris yaitu algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi.

Algoritma kriptografi yang akan digunakan pada penelitian ini adalah algoritma simetris yaitu Rivest Code 4 (RC4). Algoritma RC4 adalah algoritma yang bersifat stream cipher dimana proses penyandiannya berorientasi pada satu bit/ byte data.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka perumusan masalah yang dirumuskan adalah :

Bagaimana membuat suatu aplikasi yang dapat mengenkripsi dan mendekripsi data secara cepat dengan menggunakan algoritma Rivest Code 4 (RC 4).

Batasan Masalah

Batasan masalah dari penelitian ini antara laian:

1. Aplikasi ini dibuat menggunakan bahasa pemrograman PHP dan MySQL.
2. Algoritma yang digunakan adalah Rivest Code 4.
3. Untuk mempercepat enkripsi dan dekripsi file hanya dibatasi sebesar 2 MB.
4. Format file yang bisa dienkripsi dan didekripsi hanya .pdf,, .docx, xls, dan .txt.

1.4. Tujuan dan Manfaat

Tujuan dari penelitian ini adalah untuk membuat aplikasi kriptografi dalam pengamanan data.

Adapun manfaat dari pembuatan aplikasi tersebut adalah :

1. Memudahkan untuk menjaga keamanan data melalui proses enkripsi.
2. Menghindari pendurian data dengan menggunakan kriptografi supaya data tidak dapat disalahgunakan dengan orang yang tidak berhak.
3. Memaksimalkan proses enkripsi dan dekripsi secara optimal.

1.5. Metode Penelitian

Adapun metode pengumpulan data dan informasi yang digunakan adalah sebagai berikut:

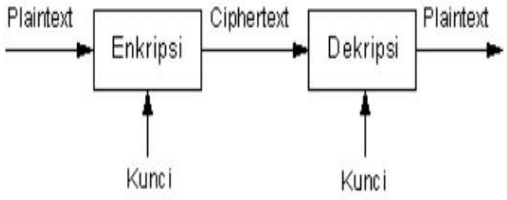
1. Analisa kebutuhan sistem
   Mendefinisikan semua kebutuhan aplikasi yang akan dibuat dengan cara mengumpulkan bahan-bahan yang didapat dari buku-buku, modul, studi lapangan.
2. Analisis dan Perancangan Aplikasi
   Perancangan aplikasi yang akan dibangun menggunakan berbagai tahap yang meliputi beberapa tahap yaitu tahap desain berdasarkan kebutuhan yang telah didefinisikan, tahap pembuatan program dan lain-lain.
3. Implementasi dan Pengujian
   Implementasi akan dilakukan sesuai dengan analisis dan perancangan yang telah dilakukan. Setelah itu, pengujian aplikasi dilakukan dengan menjalankan semua fungsi apakah berjalan sesuai yang diharapkan dalam melaksanakan tugasnya.

LANDASAN TEORI

2.1. Definisi Kriptografi

Kata kriptografi berasal dari bahasa Yunani yang terdiri dari dua buah kata yaitu *crypto* dan *graphia*. Kata *crypto* berarti *secret* (rahasia) sedangkan *graphia* berarti *writing* (tulisan).Berarti secara umum makna dari kata kriptografi adalah tulisan rahasia. Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana cara menyembunyikan pesan. Kriptografi merupakan ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi otentitas (Sadikin. 2012).

Secara umum, kriptografi merupakan teknik pengamanan informasi yang dilakukan dengan cara mengolah informasi awal (*plaintext*) dengan suatu kunci tertentu menggunakan suatu metode enkripsi tertentu sehingga menghasilkan suatu informasi baru (*ciphertext*) yang tidak dapat dibaca secara langsung. *Ciphertext* tersebut dapat dikembalikan menjadi informasi awal (*plaintext*) melalui proses dekripsi. Urutan proses kriptografi secara umum dapat dilihat pada Gambar 1.



Gambar 1. Proses Kriptografi Secara Umum

2.2. Sejarah Kriptografi

Kriptografi mempunyai sejarah yang panjang dan menakjubkan. Informasi yang lengkap mengenai sejarah kriptografi dapat ditemukan didalam buku David Khan yang berjudul *The Codebreakers*. Buku yang tebalnya 1000 halaman ini menulis secara rinci sejarah kriptografi, mulai dari penggunaan kritografi oleh bangsa Mesir 4000 tahun yang lalu (berupa *hieroglyph* pada Piramid) hingga penggunaan abad ke-20.

Sebagian besar sejarah kriptografi bagian dari kriptografi klasik, yaitu metode kriptografi yang menggunakan kertas dan pensil atau menggunakan alat bantu mekanik yang sederhana. Kriptografi klasik secara umum dapat dikelompokan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma subtitusi (*substitution cipher*). Algoritma transposisi adalah algoritma yang mengubah susunan-susunan huruf didalam pesan, sedangkan algoritma subtitusi yaitu mengganti setiap huruf atau kelompok huruf dengan sebuah huruf.-huruf lain.

Penggunaan *transposition cipher* yaitu oleh tentara Sparta di Yunani pada permulaan tahun 400 SM. Mereka menggunakan apa yang dinamakan *scytale* (Gambar 2.). *Scytale* terdiri dari sebuah kertas panjang dari daun *papyrus* yang dililitkan pada sebuah silinder dari diameter tertentu (diameter dari silinder merupakan kunci dari penyandian tersebut).Pesan ditulis baris per baris dan secara horizontal. Apabila pitadilepas, maka setiap huruf akan tersusun secara acak membentuk pesan rahasia (pesan yang tidak dapat dibaca). Agar pesan tersebut dapat dibaca, maka pesan tersebut kembali dililitkan kesilinder yang diameternya sama dengan diameter silinder pengirim (Subagja. 2015).

Gambar 2. *Scytale*

Perkembangan peralatan computer digital memicu terbentuknya kriptografi modern. Dengan computer digital, akan sangat mungkin untuk menghasilkan *cipher* yang lebih kompleks dan rumit. Kriptografi klasik pada umumnya dienkripsi karakter per karakter (menggunakan alphabet tradisional), sedangkan kriptografi modern beroperasi pada string biner *cipher* yang lebih kompleks.

Adapun tujuan kriptografi adalah sebagai berikut (Munir. 2006) :

1. *Confidentiality* (Kerahasiaan)
   Adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca.
2. *Authentiation* (Otentikasi)
   Adalah identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*). Maupun mengidentifikasi kebenaran sumber pesan.
3. *Data Integrity* (Integritas Data)
   Adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
4. *Non - repudiation* (Tanpa Penyangkalan)
   Adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan yaitu pengiriman atau penerima pesan menyangkal telah menerima pesan.

2.3. Algoritma Kriptografi

Algoritma dalam kriptografi merupakan sekumpulan aturan (fungsi matematis yang digunakan) untuk proses enkripsi dan proses dekripsi. Dalam beberapa metode kriptografi terdapat perbedaan antara fungsi enkripsi dan fungsi dekripsi.

Konsep matematis yang mendasari algoritma adalah relasi antara himpunan, yaitu relasi antara himpunan yang berisi elemen-elemen *ciphertext*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan himpunan *plaintext* dinotasikan *P* dan himpunan elemen *ciphertext* dinotasikan *C*, maka fungsi *E* memetakan himpunan *P* ke himpunan *C*.

E(P) = C

Dan fungsi dekripsi memetakan himpunan *C* ke himpunan *P*

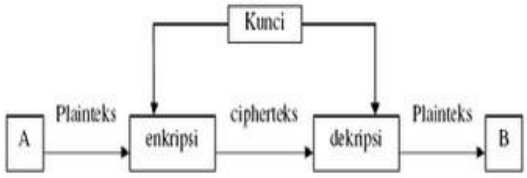
D(C) = P

Karena fungsi dekripsi D mengembalikan himpunan *C* menjadi himpunan *P* asal, maka algoritma kriptografi harus memenuhi persamaan

D(E(P)) = P

2.3.1 Algoritma Simetris

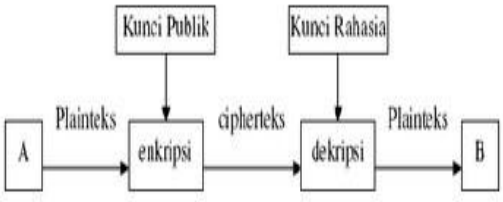
Pada umumnya yang termasuk ke dalam kriptografi simetris ini beroperasi dalam mode blok (*block cipher*), yaitu setiap kali proses enkripsi atau dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (*stream cipher*), yaitu setiap kali enkripsi atau dekripsi dilakukan terhadap satu bit atau satu byte data. Proses dari skema kriptografi simetris dapat dilihat pada Gambar 3.



Gambar 3. Algoritma Simetris

2.3.2. Algoritma Asimetris

Berbeda dengan kriptografi kunci simetris, kriptografi kunci *public* memiliki dua buah kunci yang berbeda pada proses enkripsi dan dekripsinya. Dimana kunci yang digunakan untuk proses enkripsi atau sering disebut *public key* dan dekripsi atau sering disebut *private key* menggunakan kunci yang berbeda. Entitas pengirim akan mengenkripsi dengan menggunakan kunci public, sedangkan entitas penerima mendekripsi menggunakan kunci *private*. Skema dari kriptografi dapat dilihat pada Gambar 4.



Gambar 4. Algoritma Asimetris

2.4. Algoritma Rivest Code 4 (RC 4)

RC4 merupakan salah satu jenis *stream cipher*, yaitu memproses unit atau input data pada satu saat. Dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variabel. Algoritma ini tidak harus menunggu sejumlah *input* data tertentu sebelum diproses, atau menambahkan *byte* tambahan untuk mengenkrip.

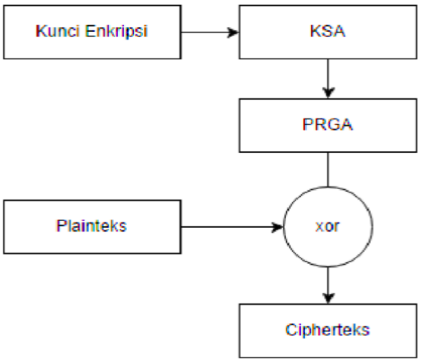
Algoritma RC4 bekerja pada 2 tahap, menyetem susunan (*key setup*) dan (*chipering*). Kunci susunan merupakan yang lebih awaldan merupakan tahap yang paling sulit dari algoritma ini. Selama menyetem susunan dari N-bit (N menjadi panjang kunci), kunci enkripsi digunakan untuk menghasilkan suatu variabel enkripsi yang menggunakan dua *arrays, state* dan *key* dan jumlah N dari operasi pencampuran. Operasi pencampuran terdiri dari menukar *bytes,* modulo operasi, dan rumusan lain. Suatu modulo operasi adalah proses sisa dari suatu hasil divisi. Sebagai contoh, 10/4 adalah 2 sisa 2, oleh karena itu 10 mod 4 sama dengan 2.

Sekali variabel enkripsi dihasilkan dari *key setup,* langkah selanjutnya adalah masuk ke fase *chipering,* di mana dalam proses ini hasilnya akan diXORkan dengan *plaintext.* Sekali penerima mendapat pesanyang dienkripsi, langkah selanjutnya adalah mendeskripsikannya dengan

mengXORkan pesan yang dienkripsi, dengan menggunakan variabel yang sama (Pengantar Ilmu Kriptografi, 2008:43)

2.4.1. Algoritma Enkripsi RC 4

RC4 merupakan jenis *stream cipher* yang mempunyai sebuah S-Box, S0,S1,…,S255, yang berisi permutasi dari bilangan 0 sampai 255, dan permutasi merupakan fungsi dari kunci dengan panjang yang variabel. Dalam algoritma enkripsi metode ini akan membangkitkan *pseudorandom byte* dari *key* yang akan dikenakan operasi XOR terhadap *plaintext* untuk menghasilkan *ciphertext*. Untuk menunjukan proses enkripsi dari algoritma RC4, berikut dapat dilihat pada Gambar 5. di bawah :



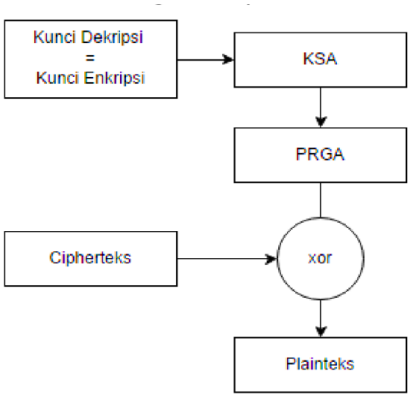
Gambar 5. Arsitektur Enkripsi RC4

2.4.2. Algoritma Dekripsi RC 4

Algoritma dekripsi RC4 mirip dengan algoritma enkripsinya, perbedaannya hanya pada saat *stream generation*, yaitu untuk menghasilkan plainteks semula, maka *ciphertext* nya akan dikenakan operasi XOR terhadap *pseudorandom* bytenya.

Algoritma key setup pada proses dekripsi sama dengan algoritma enkripsinya yang diproses inisialisasi S-Box, penyimpanan kunci kedalam *key bytearray* hingga proses inisialisasi S-Box berdasarkan *key byte array* nya. Untuk itu proses dekripsi dan enkripsi akan menghasilkan *key stream* yang sama. Perbedaannya hanya pada *stream generationnya*, yaitu yang dioperasikan bersama *key stream* adalah *ciphertext* untuk menghasilkan kembali *plaintext*.

Berikut ini akan diberikan Gambar proses dari dekripsi RC4. Lihat Gambar 6.



Gambar 6. Arsitektur Dekripsi RC4

2.5. Algoritma Huffman

Algoritma Huffman adalah salah satu algoritma kompresi. Algoritma huffman merupakan algoritma yang paling terkenal untuk mengompres teks. Terdapat tiga fase dalam menggunakan algoritma Huffman untuk mengompres sebuah teks, pertama adalah fase pembentukan pohon Huffman, kedua fase encoding dan ketiga fase decoding.

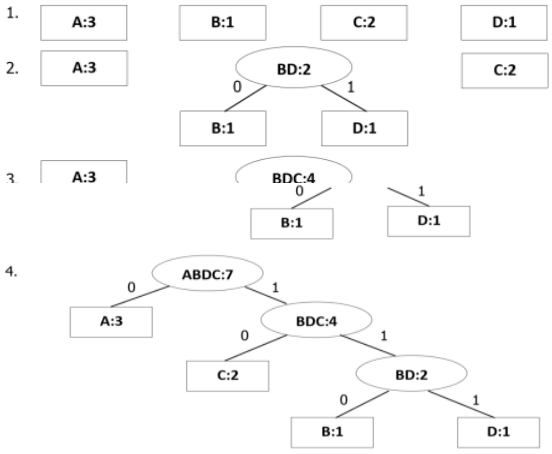
2.5.1. Algoritma Kompresi Huffman

Algoritma Huffman ditemukan oleh seorang mahasiswa MIT pada tahun 1952 bernama David Huffman. Algoritma ini termasuk dalam metode kompresstatistic yang memanfaatkan perhitungan statistika (*Statistical Methods*) untuk melihat probabilitas kemunculan data dari sebuah dokumen. Probabilitas tersebut digunakan untuk menentukan cara untuk mengolah data terebut agar bisa dipadatkan.

2.5.2. Pembentukan Pohon Huffman

Kode Huffman pada dasarnya merupakan kode prefik (*prefix code*). Kode *prefix* adalah himpunan yang berisi sekumpulan kode biner,dimana pada kode *prefix* ini tidak ada kode biner yang menjadi awal bagi kode biner yang lain. Kode *prefix* biasanya dipresentasikan sebagai pohon biner yang diberikan nilai atau label. Untuk cabang kiri pada pohon biner diberi label 0, sedangkan cabang kanan pada pohon biner diberi label 1. Rangkaian bit yang terbentuk pada setiap lintasan dari akar ke daun merupakan kode prefix untuk karakter yang berpadanan. Pohon biner ini biasa disebut pohon Huffman.

Sebagai contoh, dalam kode ASCII *string* 7 huruf “ABACCDA” membutuhkan representasi 7 x 8 bit = 56 bit (7 *byte*), dengan rincian sebagai berikut : Pada string di atas, frekuensi kemunculan A = 3, B = 1, C = 2, dan D = 1,



Gambar 7. Pohon Huffman untuk Karakter “ABACCDA”

2.2.1 Proses Encoding

Encoding adalah cara menyusun *string* biner dari teks yang ada. Proses *encoding* untuk satu karakter dimulai dengan membuat pohon Huffman terlebih dahulu. Setelah itu, kode untuk satu karakter dibuat dengan menyusun nama *string* biner yang dibaca dari akar sampai ke daun pohon Huffman.

Sebagai contoh kita dapat melihat tabel dibawah ini, yang merupakan hasil encoding untuk pohon Huffman pada tabel 1.

Tabel 1. Kode Huffman untuk Karakter "ABCD"

Karakter String	Biner Huffman
A	0
B	110
C	10
D	111

2.5.4 Proses Decoding

Decoding merupakan kebalikan dari *encoding*. *Decoding* berarti menyusun kembali data dari *string* biner menjadi sebuah karakter kembali. *Decoding* dapat dilakukan dengan dua cara, yang pertamadengan menggunakan pohon Huffman dan yang kedua dengan menggunakan tabel kode Huffman.

2.6. PHP

PHP adalah singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan web yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan web dapat dibuat dinamis sehingga maintenance situs web tersebut menjadi lebih mudah dan efisien (Nugroho:2009).

2.7. XAMPP

XAMPP adalah sebuah paket web server yang gratis dan open source cross platform yang didalamnya terdapat Apache HTTP Server, MySQL Database dan interpreter untuk script yang ditulis dalam Bahasa Pemograman PHP dan Perl. (Nugroho:2009).

2.8. MySQL

MySQL merupakan software database yang paling populer dikarenakan performa query dari database yang bisa dikatakan paling cepat, dan bisa dibilang jarang bermasalah. Mulai dari versi 3.23 MySQL menjadi software open source yang berarti gratis, dapat digunakan untuk kepentingan komersial atau personal. MySQL kini dapat digunakan di Windows, yang pada awalnya digunakan di Linux (Nugroho:2009).

2.9. Pengujian Black Box

Black box testing merupakan pengujian yang memungkinkan software engineer mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program (Pressman, 2005).

Pengujian black-box juga merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode white-box.

Pengujian black-box berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

ANALISIS DAN PERANCANGAN

3.1 Analisa Masalah

Dokumen merupakan data yang sangat penting baik itu berupa dokumen pribadi, perusahaan atau organisasi dan lain sebagainya. Oleh karena itu, sebuah dokumen seharusnya dijaga kerahasiaannya agar tidak disalahgunakan oleh orang yang tidak berhak. Disini seringkali masalah keamanan menjadi urutan kedua atau bahkan urutan yang terakhir dalam daftar hal-hal yang dianggap penting.

Apabila menggangu performa sistem, masalah keamanan ini sering dikurangi atau bahkan dihilangkan. Salah satu cara untuk mengamankan sebuah dokumen yaitu dengan mengubah dokumen asli menjadi dokumen yang tidak bisa dibaca oleh orang lain atau sering disebut dengan enkripsi.

3.2 Penyelesaian Masalah

Untuk memecahkan masalah diatas, maka dibuatlah suatu aplikasi yang dapat menjaga kerahasiaan dari sebuah dokumen atau data. Aplikasi tersebut nantinya dapat mengubah sebuah file dokumen menjadi file yang isinya tidak bisa dibaca dan dokumen tersebut terjaga kerahasiaannya. Lalu, untuk mengefisienkan penyimpanan, dokumen tersebut akan dikompresi. Kemudian mengembalikan dokumen tersebut menjadi seperti semula tanpa mengalami perubahan sedikitpun.

Dengan adanya aplikasi ini diharapkan suatu dokumen atau data penting dapat disimpan dan dikirim ke pihak yang benar-benar berhak dan tidak disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab.

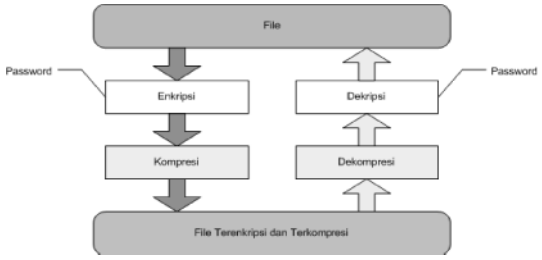
3.3 Analisa Kebutuhan Sistem

Adapun analisa kebutuhan sistem adalah sebagai berikut :

1. Aplikasi dapat memberikan fungsi otentifikasi *user* melalui proses *login.*
2. Aplikasi dapat memberikan layanan proses enkripsi (pengacakan isi data).
3. Aplikasi dapat memberikan layanan proses dekripsi (mengembalikan isi data seperti semula).
4. Aplikasi dapat memberikan layanan kompresi (pemampatan data).
5. Aplikasi dapat memberikan layanan *download file* jika telah melakukan proses enkripsi ataupun dekripsi.

3.4 Rancangan Sistem

Secara umum, rancangan program yang akan dibuat dapat dilihat pada gambar 8.



Gambar 8. Arsitektur Kerja Aplikasi

3.4.1 Rancangan Basis Data

Berikut adalah struktur tabel rancangan basis data yang terdapat pada aplikasi enkripsi dan dekripsi.

Tabel 2. Tabel Login

Kd_user	Username	Password	Level	Counter	Join_date
PK					

Tabel 3. Tabel File

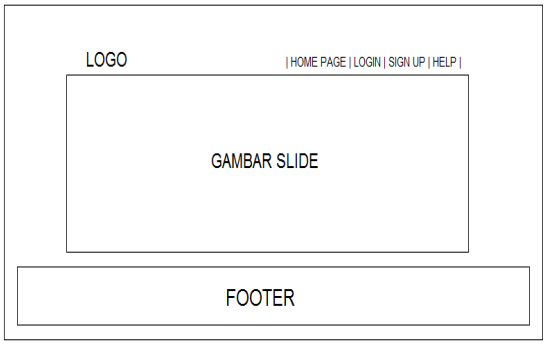
Kd_file	Nama_file	Password	Tanggal_file	Kd_user
PK				

3.4.2 Rancangan Layar

Rancangan layar sangat penting dalam membuat suatu program. Oleh karena itu rancangan layar harus mudah dimengerti dan dipahami, agar dalam menggunakan program *user* merasa nyaman dalam mengunakannya sehingga rancangan layar tidak membuat bingung *user* dan tidak mengalami kesulitan saat menggunakan program, ini.

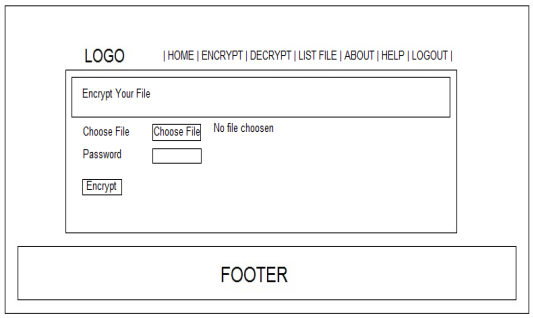
Dalam program ini, akan digambarkan rancangan layar masing-masing *form,* yaitu rancangan layar *form home*, *register, login, encrypt*, *decrypt*, *list file*, *user*, *help* dan *about.* Serta menu *logout* untuk keluar dari menu utama.

Rancangan layar pada Gambar 9 terdapat empat menu. Menu yang pertama adalah menu *Home*, yang berisi halaman utama saat aplikasi dibuka. Menu yang kedua adalah Login, dimana *user* harus *Login* terlebih dahulu untuk menggunakan aplikasi. Menu yang ketiga adalah menu Register, dimana *user* dapat mendaftar agar dapat menggunakan aplikasi. Menu yang terakhir adalah menu *Help,* dimana terdapat panduan untuk menggunakan aplikasi untuk *user.* Lihat Gambar 9.



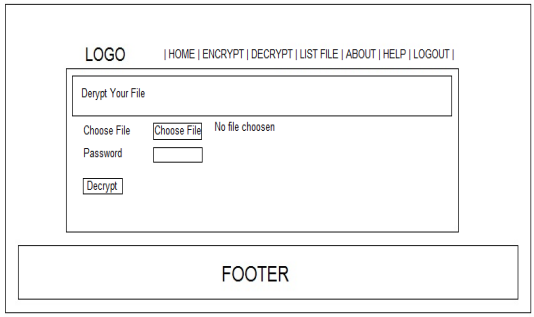
Gambar 9. Rancangan Layar *Form* Menu *Home*

Gambar 10 berikut adalah rancangan layar pada *Form Encrypt.* Untuk mengenkripsi *file,* user terlebih dahulu memilih *file* yang akan dienkripsi. Kemudian *user* harus memasukan password minimal 8 karakter agar *file* dapat dienkripsi.



Gambar 10. Rancangan Layar *Form Encrypt File*

Gambar 11 untuk mendekripsi *file user* terlebih dahulu memilih *file* yang telah dienkripsi sebelumnya. *User* harus memasukan *password* yang sama ketika melakukan proses enkripsi sebelumnya. Kemudian proses dekripsi *file* dapat dijalankan.

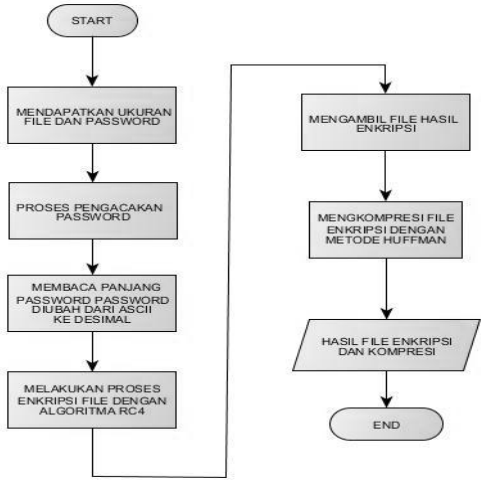


Gambar 11. Rancangan Layar *Form Decrypt File*

3.5 Flowchart

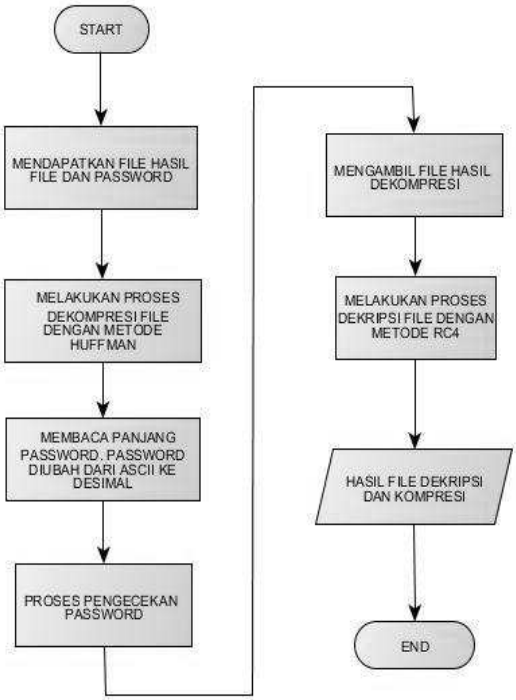
Berikut ini adalah flowchart yang digunakan untuk menelusuri proses program pada aplikasi kriptografi untuk keamanan *file*.

Flowchart ini merupakan alur jalannya proses enkripsi sebuah *file* yang ingin dienkripsi. *Flowchart* proses enkripsi dapat dilihat seperti Gambar 12.



Gambar 12. *Flowchart* Proses Enkripsi

Gambar 13 merupakan *flowchart* proses dekripsi. *Flowchart* ini menjelaskan proses pengembalian data dari *file* enkripsi menjadi *file* asli atau orisinil.



Gambar 13. *Flowchart* Proses Dekripsi

Input untuk sistem kompresi dalam algoritma kompresi data Huffman ini berupa sebuah berkas (*file*). Gambar 14. menunjukan skema kompresi dalam algoritma kompresi Huffman.

Gambar 14. *Flowchart* Proses Kompresi Huffman

Input untuk sistem dekompresi dalam algoritma kompresi data Huffman ini berupa sebuah berkas (*file*). Gambar 15. menunjukan skema dekompresi algoritma data Huffman.

Gambar 15. *Flowchart* Proses Dekompresi Huffman

IMPLEMENTASI DAN UJI COBA

4.1 Implementasi Program

Agar aplikasi enkripsi dan dekripsi ini dapat berjalan dengan baik dan bekerja sesuai dengan apa yang diharapkan, spesifikasi perangkat keras dan perangkat lunak yang dipakai untuk implementasi aplikasi ini juga harus mendukung. Berikut spesifikasi yang bisa mendukung implementasi ini, diantaranya adalah :

4.1.1 Perangkat Keras

Perangkat keras (hardware) yang dipakai untuk implementasi aplikasi ini adalah sebagai berikut :

1. Processor :Intel (R) Celeron (R) CPU 1007 @ 1.5 GHz.
2. Memory :DDR3 2 GB.
3. Monitor: LCD 17” (1366 x 768) (32-bit) (60Hz).
4. Penyimpanan: 500 GB.
5. Keyboard dan Mouse

4.1.2 Perangkat Lunak

Perangkat lunak (software) yang dipakai untuk mengimplementasi aplikasi ini adalah sebagai berikut :

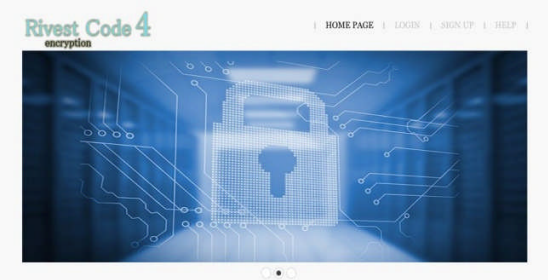
1. Sistem Operasi Windows 7 (64 Bit)
2. Notepad++
3. XAMPP 1.8.1
4. Google Chrome versi 47.0
5. MySQL Front 5.1

4.2 Implementasi Antar Muka

Pada bagian ini, akan diuraikan mengenai tampilan antar muka aplikasi ini mulai dari pertama kali dijalankan sampai selesai dijalankan. Berikut ini akan diberikan penjelasan dan gambar mengenai tampilan-tampilan yang ada pada aplikasi ini.

4.2.1 Tampilan Layar Home

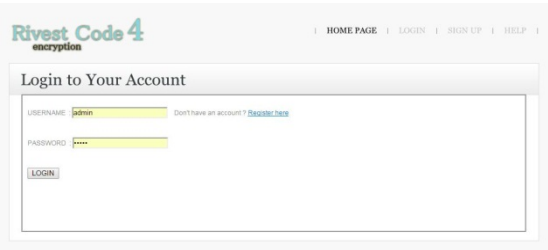
Tampilan layar dari form home pada Gambar 16. ini muncul pada saat pertama kali aplikasi dijalankan.



Gambar 16. Tampilan Layar *Form Home*

4.2.2. Tampilan Layar Login

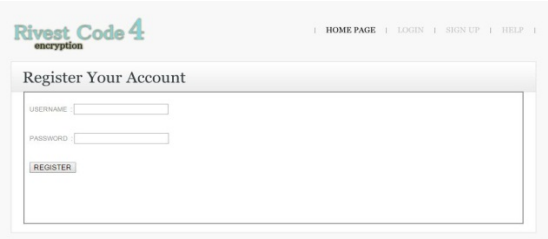
Tampilan layar dari form login pada Gambar 17 ini muncul pada saat user memilih menu login pada saat membuka aplikasi. Di dalam form login terdapat username ¬dan password. User harus memasukkan username dan password agar dapat menggunakan aplikasi ini.



Gambar 17. Tampilan Layar *Form Login*

4.2.3. Tampilan Layar Register

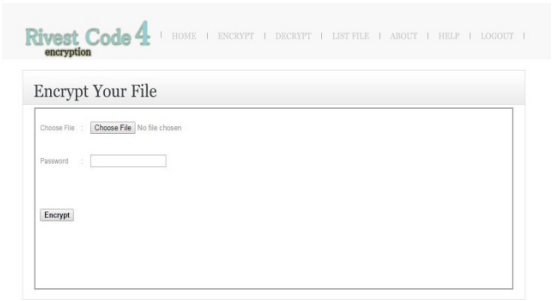
Tampilan layar dari form register pada Gambar 18 ini muncul pada saat user memilih menu register pada saat membuka aplikasi. Di dalam form register terdapat username ¬dan password. User harus mengisi username dan password untuk dapat mendaftar dan menggunakan aplikasi ini.



Gambar 18. Tampilan *Form Register*

4.2.4. Tampilan Layar Enkripsi

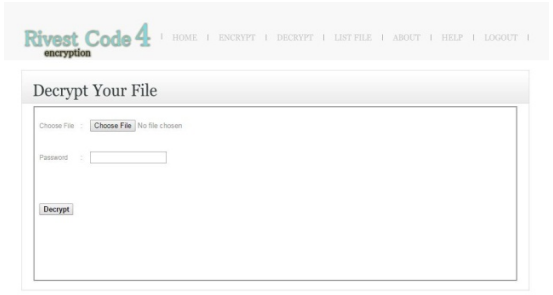
Jika user memilih menu enkripsi, maka akan muncul tampilan form enkripsi seperti Gambar 19 berikut ini.



Gambar 19. Tampilan Layar *Form* Enkripsi

4.2.5. Tampilan Layar Dekripsi

Pada form dekripsi seperti yang terlihat pada Gambar 20. berikut, user bisa melakukan proses dekripsi file yang telah dienkripsi sebelumnya.



Gambar 20. Tampilan Layar *Form* Dekripsi

4.3. Pengujian

Setelah Kebutuhan terpenuhi baik software maupun hardware, maka proses selanjutnya adalah menguji coba aplikasi yang telah dibuat. Pada bagian ini dapat diuraikan mengenai pengujian enkripsi dan dekripsi file. Pengujian tersebut nantinya akan mendapatkan hasil perbandingan file asli dan file hasil enkripsi.

4.3.1. Pengujian Black Box

Pengujian Black Box adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni aplikasi apa yang seharusnya dilakukan. Berikut ini adalah tabel pengujian Black Box pada aplikasi. Lihat tabel 4.

Tabel 4. Tabel Pengujian Black Box

No.	Skenario Pengujian	Test Case	Hasil Pengujian	Kesimpulan
1	Username atau Password salah	Username atau password salah ketika user melakukan login	Sistem akan menampilkan pesan ”Username atau Password Salah ! Mohon Login Kembali”	Valid
2	Username sudah ada ketika Register	Username sudah digunakan ketika user registrasi	Sistem akan menampilkan pesan ”Username sudah digunakan”	Valid
3	Password kurang dari 8 atau kosong ketika mengenkrip file	User memasukan password kurang dari 8 ketika mengenkrip	Sistem akan menampilkan pesan ”Password kurang dari 8 atau kosong”	Valid
4	Password kurang dari 8 atau kosong ketika mendekrip file	User memasukan password kurang dari 8 ketika mendekrip	Sistem akan menampilkan pesan ”Password kurang dari 8 atau kosong”	Valid
5	Tidak ada file yang diupload	User tidak menguploud file ketika enkrip dan dekrip	Sistem akan menampilkan pesan ”Tidak ada file yang diupload”	Valid
6	File yang diupload bukan file .docx, .pdf, .xls atau .txt ketika ingin mengenkrip file	User mengupload file format yang selain .pdf, .docx, .txt	Sistem akan menampilkan pesan ”File yang dipilih bukan file .docx, .pdf, .xls atau .txt”	Valid
7	File yang diupload bukan file hasil enkripsi ketika ingin mendekrip file	User menguploud file yang bernama selain ”Encrypt_angka_namafie”	Sistem akan menampilkan pesan ”Tidak ada file enkrip yang diupload”	Valid
8	User menghapus list file yang ada di menu list file	User menghapus file	Sistem akan menampilkan pesan ”Data file berhasil dihapus”	Valid

4.3.2. Hasil Pengujian Enkripsi dan Dekripsi

Dalam pengujian kali ini akan dibahas perbandingan antara proses enkripsi dan dekripsi atara file Xls, Doc dan file txt. Pengujiannya yaitu meliputi ukuran awal file yang ingin dienkripsi, panjang password yang digunakan, waktu proses enkripsi, waktu proses dekripsi dan hasil yang dicapai dalam proses enkripsi maupun dekripsi.

Tabel 5. Hasil Pengujian Proses Enkripsi

Input File	Password	Ukuran *File Input*	Waktu Enkripsi (Second)	Ukuran Hasil Enkripsi & kompresi	*Output File*
Latihan.xls	12345678	30 KB	2.0	13 KB	Enkrip_6876_Latihan.Xls
Latihan.doc	12345678	24 KB	1.3	12 KB	Enkrip_4180_Latihan.Doc
Soal_IJK.txt	12345678	6 KB	0.6	4 KB	Enkrip_7386_Soal_IJK.txt

Tabel 6. Hasil Pengujian Proses Dekripsi

Input File	Password	Ukuran *File Input*	Waktu Dekripsi (Second)	Ukuran Hasil Dekripsi	*Output File*
Enkrip_2821_Latihan.Xls	12345678	13 KB	1.9	30 KB	Dekrip_6876_Latihan.xls
Enkrip_4180_Bug	12345678	12 KB	2.4	24 KB	Dekrip_4180_ Bug.
Enkrip_7386_Soal_IJK.txt	12345678	4 KB	0.28	6 KB	Dekrip_7386_Soal_IJK.txt

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan perumusan masalah yang ada, maka dapat diambil suatu kesimpulan antara lain :

Aplikasi kriptografi dengan algoritma RC 4 telah berhasil direalisasikan melalui tahapan-tahapan perancangan dan pembuatan aplikasi dengan pemodelan UML.

Password enkripsi file tidak mudah terbaca, karena RC 4 melakukan pengacakan kunci yang sangat rumit sehingga sulit untuk tertembus.

Aplikasi ini telah memenuhi komponen kriptografi yaitu kerahasiaan, keutuhan, dan keaslian data.

5.2. Saran

Adapun saran yang mungkin diperlukan untuk pengembangan lebih lanjut antara lain adalah :

1 Aplikasi diharapkan bisa dikembangkan menjadi lebih baik, sehingga bisa mengenkrip format file video, musik, dan lain-lain.
2 Dikembangkan dengan algoritma yang lebih baik sehingga kecepatan dalam mengenkripsi dan mendekripsi file menjadi lebih cepat dan lebih aman.
3 Dikembangkan menggunakan algoritma kompresi yang lebih baik, agar ukuran file hasil enkripsi diharapkan dapat menjadi lebih kecil lagi.

DAFTAR PUSTAKA

[1.] Abraham, O. dan Shefiu, G.O. 2012. An Improved Caesar Cipher (ICC) Algorithm, International Journal of Engineering Science & Advanced Technology, Volume 2, Issue-5.: 1199-1202

[2.] Ariyus, Dony. 2008. Pengantar Ilmu Kriptografi : Teori Analisis & Implementasi. Yogayakarta : Andi

[3.] Basharat, I., Azam, F., & Muzaffar, A.W. 2012. Database Security and Encryption: A Survey Study. International Journal of Computer Applications (0975 – 888). Volume 47– No.12, June 2012. National University of Sciences and Technology (NUST), H-12.

[4.] Dharwiyanti dan Wahono. 2013. Pengantar Unified Modelling Language (UML). Copyright ©2003

[5.] Fowler, Martin.2005. UML Distilled. Edisi 3. Yogyakarta : Andi

[6.] Goyal, D dan Srivastava, V. 2012. Symmetric Key Algoritm, International Journal of Information and Communication Technology Research, Volume 2 No. 4.

[7.] Jogiyanto. 2005. Analisis dan Desain Sistem Informasi. Yogyakarta : Andi

[8.] Munir, Rinaldi. 2006. Kriptografi. Bandung : Informatika

[9.] Nugroho, Bunafit. 2009. Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL. Yogyakarta : Gava Media

[10.] Pressman, Roger S. 2002. Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku I), Yogyakarta : Andi

[11.] Pressman, Roger S. 2005. Software Engineering : A Practitioner's Approach?, Edisi ke 6, New York : McGraw-Hill

[12.] Sadikin, Rifki. 2012. Kriptografi Untuk Keamanan Jaringan. Yogyakarta : Andi

[13.] Subagja, Adhitia. 2015 : Scytale & Mesin Enigma, Dalam Buku, Khan, David, The Code Breakers, Saudi Arabia.