IMPLEMENTASI METODE RAIL FENCE CHIPER DAN ROW TRANSPOSITION CHIPER PADA MATA KULIAH KRIPTOGRAFI

Rusmala[1], Dianradika Prasti[2]
rusmalaoddang@yahoo.com[1], dd.prasty@yahoo.co.id[2]
Universitas Cokroaminoto Palopo[1,2]

ABSTRAK

Kriptografi merupakan salah satu mata kuliah yang ada dikurikulum program studi teknik informatika fakultas teknik komputer yang berjalan pada semester genap. Seperti yang diketahui bersama bahwa kriptografi adalah sebuah metode pengamanan yang digunakan untuk mengirimkan pesan kepada seseorang dimana pesan tersebut hanya dapat dibuka atau dibaca oleh orang tertentu. Tujuan penelitian yang dilakukan adalah untuk mengimplementasikan metode Rail Fence Chiper dan Row Transposition Chiper pada mata kuliah kriptografi. Selain itu untuk membantu dan memudahkan para tenaga pengajar yang mengampuh mata kuliah kriptografi dalam hal pemberian contoh kasus dimana aplikasi yang dirancang ini hanya difokuskan pada implementasi dua metode pengamanan yaitu Rail Fence Chiper dan Row Transposition Chiper.

Kata Kunci: Implementasi, Rail Fence Chiper, Row Transposition Chiper, Kriptografi.

I. PENDAHULUAN

I.1. Latar Belakang

Kriptografi merupakan sebuah teknologi yang memusatkan pada pengamanan data atau informasi yang dikirim. Kriptografi banyak digunakan untuk mengirim data yang tidak seharusnya dibaca oleh orang lain selain pengirim dan penerima.

Dewasa ini, sehubungan dengan perkembangan teknologi informasi yang semakin maju mengakibatkan akses informasi dan pengiriman data semakin meningkat. Ditengah-tengah peningkatan tersebut banyak oknum yang tidak bertanggung jawab kemudian mencari celah keamanan dalam jaringan untuk mengacaukan sirkulasi data dan peredaran data di internet. Hal tersebut yang kemudian mendorong perlunya digunakan teknologi kriptografi.

Ada banyak sekali jenis kriptografi yang dikenal saat ini salah satunya adalah kriptografi klasik. Kriptografi klasik memiliki metode diantaranya substitusi dan permutasi. Kedua metode ini dapat dimanfaatkan dalam membangun sistem keamanan kriptografi yang baik.

Pengajaran mata kuliah kriptografi pada teknik informatika hamper belum ada ditemui menggunakan aplikasi yang khusus dalam mempraktekkan mata kuliah kriptografi, pengajar hanya bias mengimplementasikan melalui tulisan dan belum menggunakan aplikasi.

I.2. Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dalam penelitian ini adalah bagaimana merancang dan implementasi Metode Rail Fence Chiper dan Row Transposition Chiper pada mata kuliah Kriptografi?

I.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan implementasi Metode Rail Fence Chiper dan Row Transposition Chiper pada mata kuliah kriptografi.

II. TINJAUAN PUSTAKA

II.1. Algoritma

Oktaviana, B (2013) menjelaskan Algoritma adalah prosedur langkah demi langkah untuk perhitungan. Algoritma digunakan untuk perhitungan pemrosesan data dan penalaran otomatis. Algoritma adalah metode efektif diekspresikan sebagai rangkaian terbatas dari instruksi-instruksi yang telah didefinisikan dengan baik untuk menghitung sebuah fungsi. Dimulai dari kondisi awal dan input awal (mungkin kosong), instruksi-instruksi tersebut menjelaskan sebuah komputasi yang bila

dieksekusi, diproses lewat sejumlah urutan kondisi terbatas yang terdefinisi dengan baik, yang pada akhirnya menghasilkan "keluaran" dan berhenti di kondisi akhir. Transisi dari satu kondisi ke kondisi selanjutnya tidak harus deterministik, beberapa algoritma, dikenal dengan algoritma pengacakan, menggunakan masukan acak.

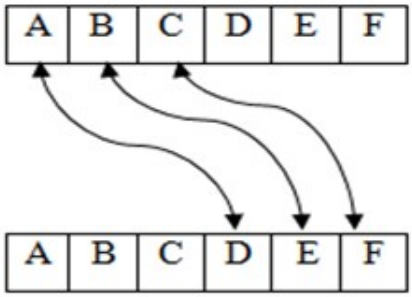
II.2. Kriptografi

Uzzin, Isbat (2006) menjelaskan Kriptografi merupakan keahlian dan ilmu dari cara-cara untuk komunikasi aman pada kehadirannya di pihak ketiga. Secara umum, kriptografi ialah mengenai mengkonstruksi dan menganalisis protokol komunikasi yang dapat memblokir lawan; berbagai aspek dalam keamanan informasi seperti data rahasia, integritas data, autentikasi, dan non-repudansi merupakan pusat dari kriptografi modern. Kriptografi modern terjadi karena terdapat titik temu antara disiplin ilmu matematika, ilmu komputer, dan teknik elektro. Aplikasi dari kriptografi termasuk ATM, *password* komputer, dan *E-commerce*.

Standar Enkripsi Data (SED) dan Standar Enkripsi Lanjutan (SEL) merupakan desain *chiper block* yang telah ditunjuk sebagai standar kriptografi oleh pemerintah Amerika (walaupun penunjukan SED pada akhirnya ditarik setelah SEL diadopsi). Walaupun penarikannya sebagai standar resmi, SED (masih menjadi varian yang masih terbukti dan lebih aman) masih cukup terkenal. Hal ini digunakan oleh banyak penerapan dari enskripsi ATM hingga keamanan *e-mail* dan akses *remote* aman. Banyak *chiper block* lainnya telah didesain dan dirilis, dengan kualitas yang bervariasi. Banyak telah juga yang dihancurkan, seperti FEAL.

II.3. Metode Substitusi

Kurniawan, Yusuf (2004) mengatakan bahwa Dalam Kriptogram *caesar cipher* dikenal dengan beberapa nama seperti *cipher*, *caesar's code* atau *caesar shift*. *Caesar chiper* merupakan teknik enkripsi yang paling sederhana dan banyak digunakan.*Cipher* ini berjenis *cipher* subtitusi, dalam setiap huruf pada *plaintext*-nya digantikan dengan huruf lain yang tetap pada posisi alphabet, misalnya diketahui bahwa pergeseran=3, maka huruf A akan digantikan oleh huruf D huruf B menjadi huruf E,dan seterusnya.



Gambar 1. Teknik Enkripsi yang Sederhana

Cara kerja sandi ini dapat diilustrasikan dengan membariskan dua set *alphabet-alphabet* sandi disusun dengan cara menggeser *alphabet* biasa ke kanan atau ke kiri dengan angka tertentu (angka ini disebut kunci), misalnya sandi *Caesar* dengan kunci 3 adalah sebagai berikut:

Alphabet biasa :ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alphabet sandi :DEFGHIJKLMNOPQRSTUVWXYZABC

Untuk menjadikan sebuah pesan,cukup mencari setiap huruf yang hendak disandikan di alphabet biasa,lalu tuliskan huruf yang sesuai pada alphabet sandi. Untuk memecahkan sandi tersebut gunakan cara sebaliknya. Contoh penyandian sebuah pesan sebagai berikut;

Teks terang : kirim pasukan kesayap kiri
Teks tersandi : NLULP SDVX NDQ NH VDBDS NLUL

Pada zaman dulu sandi *caesar* ini terbukti cukup aman untuk melindungi pesan-pesan rahasia, mungkin salah satunya karena musuh-musuhnya Julius Caesar mengira bahwa pesan tersandi itu merupakan bahasa asing lain yang tidak mereka ketahui.

Pemecahan terhadap sandi *caesar* ini mulai dapat dilakukan setelah pada abat ke-9 Masehi seorang ilmuan Arab, yang bernama Al-Kindi menemukan yang namanya analisis frekuensi kemunculan huruf-huruf tertentu, secara matematis sandi *caesar* ini dapat ditulis dengan

$E_n(x) = (x + n) \bmod 26$

Dn (x) = (x + n) mod 26
Dimana A = O.B =1, 2, 3,…, 25
n = nilai pergeseran
x = huruf yang akan disandi/dibuka

II.4. Metode Permutasi

Algoritma kriptografi rantai segitiga merupakan algoritma yang dibuat guna memperbaiki algoritma kriptografi klasik khususnya algoritma substitusi abjad tunggal yang sangat mudah diserang dengan teknik analisis frekuensi.

Algoritma kriptografi rantai segitiga merupakan *cipher* yang ide awalnya dari algoritma kriptografi *One Time Pad*, yaitu kunci yang dibangkitkan secara random dan panjang kunci sepanjang *plaintext* yang akan dienkripsi. Tetapi pada algoritma kriptogarfi rantai segitiga pembangkitan kunci-kunci tersebut secara otomatis dengan teknik berantai.

Berikut gambaran sederhana *cipher* rantai segitiga: Misal *plaintext* yang akan dienkripsi adalah "ABCDEFG" dengan kunci 3 dan deret bilangan pengali adalah bilangan asli.

Plainteks : ABCDEFG
1 X 3 DEFGHIJ
2 X 3 KMNOP
L3 X 3 UVWXY
4 X 3 HIJK
5 X 3 XZ
Y6 X 3 QR
7 X 3 M

Maka *ciphertext* yang dihasilkan DKUHXQM

Contoh diatas merupakan gambaran sederhana *cipher* rantai segitiga yang beroperasi pada alfabet 26 karakter. Angka disebelah kiri deret karakter merupakan nilai pergeseran. Nilai pergeseran berfungsi untuk menggeser huruf-huruf yang mekanismenya sama seperti pada *caesar cipher*,yaitu untuk operasi pada alfabet 26 karakter, sehingga pada proses enkripsi tiap barisnya berlaku :

ci=E(pi)=(pi+(k*R[baris]))mod 26

ci adalah karakter ke-i *ciphertext* sedangkan adalah karakter ke-i *plaintext*. R adalah tabel yang berisi pola faktor pengali. Maka untuk bilangan asli R[1]=1, R[2] = 2, R[3]= 3, dan seterusnya. Jika faktor pengali bilangan ganjil maka R[1]=1, R[2]=3, R[3]= 5, dan seterusnya. Jika faktor pengali adalah bilangan dengan pola menyesuaikan (*costumize*), maka nilai tabel R pun berisi nilai-nilai tersebut.

Untuk kasus di atas maka untuk baris ke-1 yaitu A = 0, B = 1 , C = 2 , dan seterusnya sesuai dengan gambar 7 maka :

C1 =(0 +(3*1)) mod 26
= 3 (D)
C2 =(1 +(3*1)) mod 26
= 4 (E)
C3 =(2 +(3*1)) mod 26
= 5 (F)

C4 ... C7 mekanismenya sejenis seperti 3 contoh diatas.

II.5. Teknik Permutasi (*Transporisi Chiper*)

Rinaldi dan Murni (2006) menjelaskan bahwa teknik dari permutasi ini juga dikenal dengan menggunakan permutasi karakter,yang mana dengan menggunakan teknik ini pesan yang asli tidak dibaca kecuali memiliki kuncu untuk mengembalikan pesan tersebut kebentuk semula atau disebut dengan deskripsi. Sebagai contoh: Ada 6 kunci untuk yang digunakan untuk melakukan permutasi chipper .

Tabel 1. Kunci Permutasi Chipper

1	2	3	4	5	6
3	5	1	6	4	2

Dan 6 kunci untuk inverse dari permutasi tersebut seperti pada tabel.

Tabel 2. Kunci Inverse dari Permutasi

1	2	3	4	5	6
3	6	1	5	2	4

II.6. Konsep Perancangan UML

Nugroho Adi (2009) menjelaskan bahwa *Unified Modelling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML sesuai dengan kata terakhir dari kepanjangan, UML itu adalah salah satu bentuk *language* atau bahasa. Menurut pencetusnya, UML di definisikan sebagai

bahasa visual untuk menjelaskan, memberikan spesifikasi sistem. UML bisa digunakan untuk :

1. Menggunakan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan aktor.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan interaction diagram.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagram*.
5. Menyatakan arsitektur implementasi fisik menggunakan komponen dan *development diagram*.
6. Menyampaikan atau memperluas fungsionality dengan *stereotypes*.

Use case merupakan pemodelah untuk menggambarkan kelakuan (*behavior*) sistem yang dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat, berikut adalah simbol-simbol yang ada pada *use case diagram*.

Tabel 3. *Use Case Diagram*

SIMBOL	NAMA	KETERANGAN
	Aktor	Menspesipikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*)akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Generalization	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk(*ancestor*)
	Include	Menspresikan bahwa use case sumber secara *eksplisit*
	Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan
	Association	Apa yang menghubungkan antara objek satu dengan yang lain
	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas
	Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah elemen-elemennya(sinergi)
	Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber: Rossa dan Shalahuddin (2015)

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut metode atau operasi:

1. Attribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Attribut mendeskripsikan property dengan sebaris teks didalam kotak kelas tersebut
3. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

II.7. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas .NET Framework). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi *Windows Mobile* (yang berjalan di atas .NET Compact Framework).

Visual Studio kini telah menginjak versi *Visual Studio 9.0.21022.08*, atau dikenal dengan sebutan *Microsoft Visual Studio 2008* yang diluncurkan pada 19 November2007, yang ditujukan untuk platform *Microsoft .NET Framework 3.5*. Versi sebelumnya, *Visual Studio 2005* ditujukan untuk platform *.NET Framework 2.0 dan 3.0. Visual Studio 2003* ditujukan untuk *.NET Framework 1.1,* dan *Visual Studio 2002* ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul *Visual*

Studio .NET, terdapat *Microsoft Visual Studio 6.0 (VS1998)*.

Gambar 2. Tampilan Microsoft C#2010

II.8. Fungsi Pengujian Whitebox Testing

Pengujian Kotak Putih (*Whitebox Testing*) merupakan pengujian yang memerlukan pemeriksaan yang detail dan prosedural. Rangkaian logika dari software diujicoba dengan menyediakan objek ujicoba yang memerlukan sekumpulan dari suatu kondisi dan perulangan tertentu.Status program dapat diperiksa dari beberapa titik secara bervariasi untuk menentukan apakah status yang diharapkan atau ditegaskan sesuai dengan status sesungguhnya.

Pengertian dan Fungsi Pengujian Kotak Putih (*Whitebox Testing*) merupakan materi yang terpenting, pada dasarnya Pengujian Kotak Putih (*Whitebox Testing*) dilakukan oleh developer yang sudah mengerti tentang isi secara menyeluruh sistem, dan mereka harus menemukan kesalahan-keselahan dalam sistem tersebut.

Untuk mengetahui fungsi dari pengujian kotak putih dengan detail, simak contoh penjelasan dibawah ini yang lebih detail, dan berikut beberapa proses yang dilakuandalam pengujian ini, yaitu :

1. Menggunakan perancangan atau kode sebagai sebuah dasar dan tergambar dalam grafik alir yang berfungsi sebagai notasi yang berguna untuk memahami aliran kontrol dan menggambarkan performa.
2. Menentukan kompleksitas siklomatik dari aliran grafik yang dihasilkan, yang berguna untuk memperkirakan modul-modul yang kemungkinan besar akan terbukti salah dan memastikan bahwa semua pernyataan telah dieksekusi minimal sekali. Kompleksitas dihitung dalam salah satu dari tiga cara berikut:
   a. Jumlah daerah-daerah grafik alir yang berhubungan dengan kompleksitas siklomat.
   b. Komplesitas siklomat V (G) untuk grafik alir G didefinisikan sebagai V(G)=E–N+2
      Dimana E adalah jumlah edge grafik alir dan N adalah jumlah node grafikalir
   c. Kompleksitas siklomatik V (G) untuk grafik aliran G juga didefinisikan sebagai V(G)=P+1
      Dimana P adalah jumlah node predikat yang terdapat dalam grafik alir G
3. Menentukan sebuah basis set dari jalur independen linier, yang berfungsi untuk memperkenalkan setidaknya satu kumpulan pernyataan-pernyataanpemrosesan atau kondisi baru.

III. METODE PENELITIAN

III.1. Jenis Penelitian

Adapun jenis penelitian yang digunakan adalah penelitian kualitatif deskriptif, menggunakan metode R&D dimana hasil penelitian nantinya mengarah ke bentuk aplikasi.

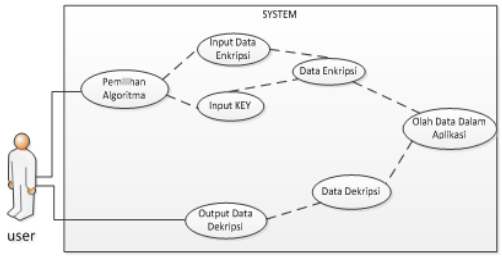
III.2. Batasan Penelitian

Ada beberapa batasan dalam penelitian ini diantaranya:

1. Fokus pada implementasi pembuatan aplikasi.
2. Aplikasi yang dibuat berdasarkan hasil rancangan dari peneliti sebelumnya.
3. Untuk perancangan menggunakan model UML, untuk listing programnya menggunakan bahasa pemrograman PHP dengan MySQL sebagai databasenya.

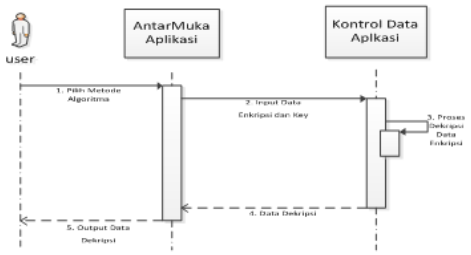
III.3. Perancangan Aplikasi

1. Use Case Diagram



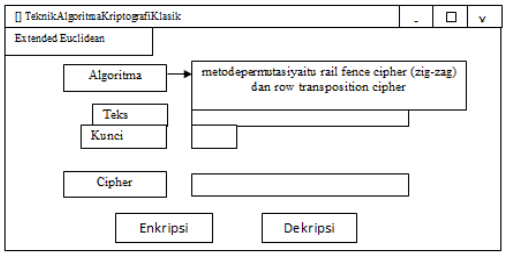
Gambar 3. *Use Case Diagram*

2. *Sequence Diagram*



Gambar 4. *Sequence Diagram Admin*

III.4. Rancangan Input dan Output



Gambar 5. Rancangan *Input* dan *Output* dari Aplikasi

Penjelasan rancangan aplikasi teknik algoritma kriptografi klasik:

1. Awal dari alur kerja aplikasi di atas adalah masuk pada tampilan menu aplikasi.
2. Pada tampilan aplikasi, akan muncul algoritma permutasi, teknik *rail fence chipper* (*zig-zag*) dan *row transposition chipper*.
3. Pilih algoritma yang akan digunakan.
4. Kemudian masukan plainteks atau teks, kunci dan kemudian enskripsi akan muncul pada *chipper*.

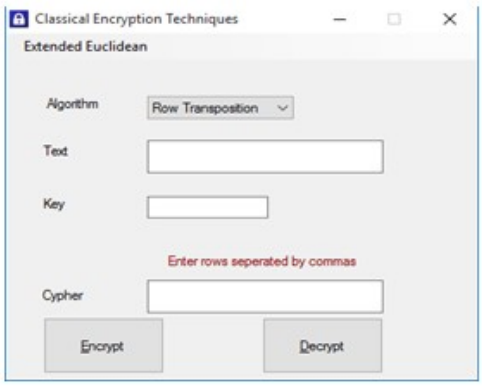
IV. HASIL DAN PEMBAHASAN

IV.1. Hasil

Berdasarkan hasil temuan dari observasi dan wawancara sebelum penelliti melakukan tahap perancangan dan pembuatan adalah sampai saat ini metode penilaian yang dilakukan oleh guru untuk melihat sampai dimana kemampuan, kualitas dari proses pembelajaran yang dilakukan selama kurang lebih satu semster terhadap siswa- siswinya di mata pelajaran yang guru ampuh dinilai dari kondisi , evaluasi . evaluasi merupakan tahap akhir yang menjadi penentu apakah yang tersusun dalam capaian silabus terpenuhi atau tidak. Evaluasi yang dilakuka ada beberapa yaitu pra tes, diskusi, ujian tengah semester dan ujian akhir sementer. Dalam penelitian ini peneliti lebih menitberatkan penelitian ke tahap evaluasi terakhir yaitu UAS. Dimana sistemnya masih menggunakan secarik kertas dan itu membutuhkan waktu yang cukup menyita bagi guru.

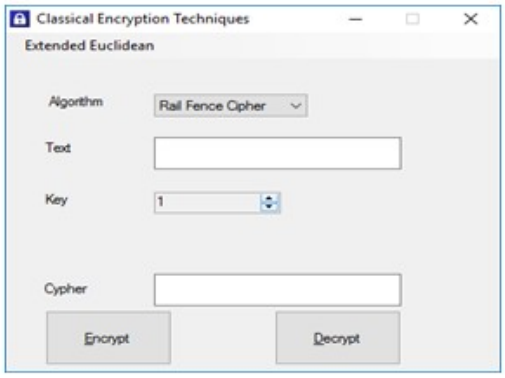
Berdasarkan informasi tersebut peneliti dapat merancang dan membuat aplikasi ujian akhir semester dengan mengunakan beberapa aplikasi dan perangkat pendukung. Hasil dari rancangan dan pembuatan tersebut adalah:

1. Form Row Transposition Chiper
   Berikut ini desain tampilan dari *row transposition chiper*.



Gambar 6. Tampilan form *Row Transposition Chiper*

2. Form Rail Fence Chiper
   Berikut ini desain tampilan dari *Rail Fence Chiper*



Gambar 7. Tampilan Form *Rail Fence Chiper*

Karena berupa aplikasi maka tampilan halaman depan, halaman *input* dan halaman *output* berada dalam satu form saja.

V. KESIMPULAN DAN SARAN

V.1. Kesimpulan

Kesimpulan yang dapat peneliti uraikan dari penelitian ini adalah ada

beberapa tahapan yang dilakukan dalam penelitian yaitu tahapan pengumpulan data, analisis data, perancangan, pembuatan serta implementasi aplikasi dimata kuliah kriptografi.

1. Tahap pengumpulan menggunakan teknik trigulasi yang terdiri dari observasi, wawancara dan tinjauan pustaka (dokumen).
2. Tahap analisis terdiri dari kebutuhan fungsional, non fungsional, pemecahan masalah.
3. Tahap perancangan terdiri dari perancangan UML (*use case*, *sequence* dan *class diagram*).
4. Tahap implementasi dalam hal ini adalah tahap pembuatan menggunakan bahasa pemrograman PHP, MySQL untuk database.

DAFTAR PUSTAKA

[1]. Kurniawan, Yusuf. 2004. Kompleksitas dan Analisis Sandi Linier Algoritma Enkripsi Substitusi Permutasi Sederhana 128 bit.Bandung.
[2]. Nugroho, Adi. 2009. Rekayasa Perangkat Lunak UML & Java. Yogyakarta.
[3]. Oktaviana, B. 2013. Kombinasi Algoritma Caesar Chiper dan Vigenere Chiper pada Three-pass protocol. Yogyakarta.
[4]. Uzzin, Isbat.2008. Diktat Kuliah Security Jaringan Introduction Kriptografi. Teknik Informatika: Politeknik Elektronika Negeri Surabaya-ITS.