IMPLEMETASI KOMBINASI ALGORITMA RAIL FENCE CIPHER DAN CAESAR CIPHER UNTUK KEAMANAN FILE

SKRIPSI

OLEH :

MERCY PRISTIAN LASTIN
NPM : 19010200P

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU
2023

IMPLEMETASI KOMBINASI ALGORITMA RAIL FENCE CIPHER DAN CAESAR CIPHER UNTUK KEAMANAN FILE

SKRIPSI

OLEH :

MERCY PRISTIAN LASTIN
NPM : 19010200P

Diajukan Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Program Studi Informatika

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU
2022

IMPLEMETASI KOMBINASI ALGORITMA RAIL FENCE CIPHER DAN CAESAR CIPHER UNTUK KEAMANAN FILE

SKRIPSI

OLEH :

MERCY PRISTIAN LASTIN
NPM : 19010700P

DISETUJUI OLEH :

Menyetujui,:

Pembimbing Utama	Pembimbing Pendamping
Siswanto, S.E, S.Kom, M.Kom	**Juju Jumadi, S.Kom, M.Kom**
NIDN. 02.240363.01	**NIDN. 02.151174.01**

Mengetahui,
Ketua Program Studi Informatika

Liza Yulianti, S.Kom, M.Kom
NIDN. 02.160772.01

IMPLEMETASI KOMBINASI ALGORITMA RAIL FENCE CIPHER

DAN CAESAR CIPHER UNTUK KEAMANAN FILE

SKRIPSI

OLEH :

MERCY PRISTIAN LASTIN
NPM : 19010200P

Telah dipertahankan di depan Tim Penguji Universitas Dehasen Bengkulu Pada :

Hari : Senin
Tanggal : 05 Juni 2023
Tempat : Ruang Sidang Universitas Dehasen Bengkulu

Skripsi Telah Diperiksa dan Disahkan Oleh :

Penguji	Nama	NIDN	Tanda Tangan
Ketua	Siswanto, SE, S.Kom, M.Kom	02.240363.01	
Anggota	Juju Jumadi, S.Kom, M.Kom	02.111282.01	
Anggota	Sapri, S.Kom, M.Kom	02.150171.02	
Anggota	Eko Suryana, S.Kom, M.Kom	02.1511740.1	

Mengetahui,

Dekan
Fakultas Ilmu Komputer

DAFTAR RIWAYAT HIDUP

Penulis bernama Mercy Pristian Lastin dilahirkan di Bengkulu tanggal 12 Agustus 1988. Anak pertama dari tiga bersaudara. Ayah bernama Lahirin dan ibu bernama Nurhayati

Menyelesaikan pendidikan sekolah dasar (SD) Negeri 12 Bengkulu Selatan pada tahun 2000. Kemudian penulis melanjutkan di sekolah menegah pertama (SLTP) Negeri 4 Bengkulu Selatan dan lulus pada tahun 2003 dan menyelesaikan pendidikan Sekolah Menengah Atas (SMA) di SMA Muhammadiyah Bengkulu Selatan pada tahun 2006 kemudian melanjut pendidikan keperguruan tinggi yaitu pada Universitad Dehasen (UNIVED) Bengkulu dengan mengambil jurusan Informatika pada Fakultas Ilmu Komputer, untuk jenjang Strata Satu (S-1).

MOTTO DAN PERSEMBAHAN

Dengan mengucapkan Syukur Alhamdulillah Atas semua nikmat yang telah Allah SWT berikan kepada saya yang pada akhirnya saya dapat menjalankan serta menyelesaikan amanah dan kewajiban saya untuk mencapai cita-cita. Saya sadar ini bukan akhir dari perjuangan tapi melainkan awal dari semua cerita untuk mencapai cita-cita dan membahagiakan orang-orang yang aku sayang. Ku persembahkan kado kecil ini dengan sepenuh hati untuk:

- Ayah dan ibuku (Aswan dan Pertiwi (ALM)) yang telah memberikan kasih sayang, memberi suport, tiada henti memberikan Do'a-Do'a, yang selalu sabar sampai saya menyelesaikan pendidikan ini. Ayah dan Alm Ibu telah melalui banyak perjuangan dan rasa sakit. Tapi saya berjanji tidak akan membiarkan semua itu sia-sia. I love you mak bak
- Istriku (Nora Natalia) terimakasih kamu selalu mendukung dan support suamimu dari awal masuk ke perguruan tinggi di Universitas Dehasen (UNIVED) sampai menyelsaikan Skripsi.
- Buat Anak-anak yang aku sayang (Ferdian Saputra, Rafael Admaja, Fina Ega Lestari, Fera Aulia) terimaksih kalian empat saudara sudah mendukung ayahmu telah menyelesaikan ke perguruan tinggi
- Kakak ku Roydi yang selalu memberikan semangat. Makasih Kakak ku sayang.
- Buat Adek ku Robi dan Roza Amelia terimakasih telah memberikan semangat selama ini.
- Buat keluarga besar terimakasih selalu ada di saat keluarga saya butuhkan (makcik, Bakcik, cicik, mangcik, makwo, bakwo, bucik dll)
- Sahabatku Agra dan Mummad Iqbal terimakasih telah membantu dalam penyelesaain skripsi ini
- Para dosen dan pembimbingku (Ibu Liza Yulianti, M.Kom dan Bapak Juju Jumadi M.Kom) yang telah membantu dalam menyelesaikan skripsi ini
- Almamater kuning yang aku banggakan.

SURAT PERNYATAAN ORISINILITAS DAN PERSETUJUAN AKADEMIK SKRIPSI

Yang bertanda tangan dibawah ini :

Nama	: Mercy Pristian Lastin
NPM	: 19010200P
Program Studi	: Informatika
Fakultas	: Ilmu Komputer

Dengan ini menyatakan dengan sesungguhnya bahwa Skripsi dengan Judul :

IMPLEMETASI KOMBINASI ALGORITMA RAIL FENCE CIPHER DAN CAESAR CIPHER UNTUK KEAMANAN FILE

1. Adalah benar dibuat oleh saya sendiri untuk memenuhi persyaratan kelulusan akademik.
2. Pada bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah ditulis sumbernya secara jelas sesuai dengan norma, kaidah dan cara penulisan ilmiah.
3. Jika dikemudian hari diketahui berdasarkan bukti-bukti yang kuat ternyata skripsi tersebut dibuat oleh orang lain atau diketahui bahwa skripsi tersebut merupakan *plagiat/mencontek/menjiblak* hasil karya tulis ilmiah orang lain, maka dengan inisaya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya sesuai dengan peraturan yang berlaku.
4. Dan atas orisinilitas tersebut diatas, maka saya menyetujui untuk memberi kepada Universitas Dehasen Bengkulu hak atas bebas royalti non-eksklusif untuk menyimpan, mengalih mediakan, mendistribusikan dan mempublikasikan skripsi saya tanpa perlu meminta izin selama mencantumkan nama saya sebagai penulis/pencipta.
5. Saya bersedia menanggung secara pribadi tanpa melibatkan Universitas Dehasen Bengkulu segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam Karya Ilmiah saya ini

Demikian surat pernyataan ini dibuat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Bengkulu, Juni 2023

Hormat Saya

Mercy Pristian Lastin

KATA PENGANTAR

Puji Syukur saya panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan karunia-NYA, sehingga skripsi yang berjudul "**Implemetasi kombinasi algoritma Rail Fence Cipher dan Caesar Cipher Untuk Keamanan File** " dapat diselesaikan dalam waktu yang telah ditetapkan.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan skripsi ini kepada :

1. Bapak Prof. Dr. Husaini, SE, M.Si, Ak, CA, CRP selaku Rektor Universitas Dehasen Bengkulu
2. Bapak Siswanto, SE, S.Kom, M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Dehasen Bengkulu sekaligus pembimbing utama yang telah membimbing dengan sabar dan memberikan masukan serta saran kepada penulis .
3. Ibu Liza Yulianti, S.Kom, M.Kom selaku Ketua Prodi Informatika Universitas Dehasen Bengkulu.
4. Bapak Juju Jumadi, S.Kom, M.Kom Selaku pembimbing pendamping telah membimbing dengan sabar dan memberikan masukan serta saran kepada penulis
5. Buat teman-teman yang tidak bisa disebutkan satu persatu baik formal dan non formal, terima kasih atas bantuannya selama penyelesaian penulisan skripsi ini.

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini, namun penulis mengharapkan saran dan kritik yang sifatnya membangun guna menunjang perkembangan ilmu pengetahuan khususnya ilmu komputer.

Bengkulu, Mei 2023

Penulis

ABSTRAK

IMPLEMETASI KOMBINASI ALGORITMA RAIL FENCE CIPHER DAN CAESAR CIPHER UNTUK KEAMANAN FILE

Oleh :

Mercy Pristian Lastin [1]
Siswanto, SE, S.Kom, M.Kom[2]
Juju Jumadi, S.Kom, M.Kom[2]

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu data, pesan dan informasi. *Myszkowski Transposition* bekerja dengan dilakukan dengan cara menyusun *plaintext* ke dalam baris matriks. Kemudian matriks tersebut dibaca perkolom sehingga didapatkan *ciphertext.* Banyak kolom yang digunakan pada matriks ditentukan oleh panjang kunci yang digunakan. Caesar Cipher melakukan pergeseran terhadap semua karakter pada plainteks dengan nilai pergeseran yang sama. Dengan mengkombinasikan algoritma Vigenere Cipher dan Caesar Cipher tersebut menghasilkan sebuah metode yang dapat memberikan tingkat keamanan yang lebih baik dibandingkan dengan penerapan masing – masing metode tersebut secara terpisah.

Hasil dari analisa dan pengujian yang dilakukan dengan menggunakan kunci dekripsi yang berbeda menghasilkan chiperteks tidak dapat dikembalikan yang mana menunjukkan hal yang normal dikarenakan metode yang digunakan merupakan kriptografi simetris sehingga proses dekripsi hanya bisa dilakukan menggunakan kunci yang sama dengan kunci pada saat dekripsi.

Kata kunci : Keamanan, Rail Fence Cipher, Ceasar Cipher,

1. Mahasiswa
2. Pembimbing

ABSTRACT

DAFTAR ISI

Halaman

BAB V KESIMPULAN DAN SARAN

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar **Halaman**

DAFTAR TABEL

Tabel **Halaman**

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesatnya perkembangan teknologi informasi menjadi salah satu alasan perlu adanya peningkatan terhadap keamanan data dalam melakukan kegiatan baik dalam bidang ekonomi, industri, kesehatan, pendidikan dan sebagainya. Kemajuan dalam melakukan komunikasi data saat ini hampir mempengaruhi semua bidang sehingga dibutuhkan teknik atau metode yang mampu untuk menjaga pesan atau data yang terkandung pada suatu informasi dari pihak-pihak yang tidak bertanggung jawab.

Keamanan merupakan sekumpulan prosedur, tahapan, dan kebijakan untuk menghentikan dan memonitoring akses tidak sah, *problem solving*, pengungkapan, gangguan pada suatu komunikasi data komputer. Pengamanan data atau pesan dapat dilakukan dengan menggunakan kriptografi. Kriptografi salah satu komponen penting untuk komunikasi dan transmisi informasi melalui layanan keamanan. Kriptografi merupakan seni maupun ilmu yang menghasilkan pesan yang rahasia (Yusrizal, 2019), selain itu kriptografi dapat diartikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya (Sumarno, dkk : 2018). Dalam kriptografi, dilakukan enkripsi untuk menyandikan *plaintext* (pesan asli) menjadi *ciphertext* (pesan tersandi) dengan mengubah pesan menjadi bentuk lain. (Ziliwu, dkk 2022).

Kriptografi terdiri dari 2 jenis yaitu kriptografi klasik dan kriptografi modern. Kriptografi klasik memiliki beberapa algoritma yang dapat digunakan untuk keamanan data diantaranya Rail Fence Cipher, Vigenere Cipher, Ceasar Cipher, Hill Cipher, Affine Cipher dan lainnya (Sutoyo & Murhaban, 2016). Dalam kriptografi *Caesar Cipher* dikenal dengan beberapa nama seperti: *shift cipher*, *Caesar's code* atau *Caesar shift*. *Caesar Cipher* merupakan teknik enkripsi yang paling sederhana dan banyak digunakan. Cipher ini berjenis cipher substitusi, dimana setiap huruf pada *plaintext*-nya digantikan dengan huruf lain yang tetap pada posisi alfabet (Sumandri, 2017), sedangkan *Rail Fence Cipher m*erupakan salah satu algoritma cipher transposisi yang mengacak urutan huruf-huruf pesan. Algoritma ini melibatkan penulisan plainteks ke bawah secara berturut turut yang memiliki baris atas dan baris bawah. Sedangkan ciphertext nya diperoleh dengan membaca huruf berdasarkan baris (Purnamasari, 2021). Algoritma *Rail Fence Cipher* menyusun plainteks secara ziq-zag dengan turun kebawah dan naik keatas sesuai ukuran kolom dan baris yang ditentukan oleh *key* (Girsang, dkk : 2019).

Berdasarkan latar belakang diatas maka penulis tertarik untuk melakukan penelitian yang dituangkan dalam bentuk proposal skripsi yang diberi judul "**Implemetasi kombinasi algoritma *Rail Fence Cipher* dan *Caesar Cipher* untuk keamanan file**".

.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang di atas, rumusan masalah yang akan dibahas adalah implementasi kombinasi algoritma *Rail Fence Cipher* dan *Ceasar Cipher* Untuk Keamanan File.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

1. Pesan berupa teks yang terdiri dari karakter ASCII 32 hingga 126
2. Panjang kunci yang digunakan maksimal sepanjang *plaintext*
3. Aplikasi dibangun menggunakan bahasa pemrograman Visual Basic.Net

1.4 Tujuan Penelitian

1.4.1 Tujuan Umum

Sebagai salah satu syarat untuk melanjutkan penulisan skripsi pada Fakultas Ilmu Komputer Program Studi informatika Universitas Dehasen Bengkulu

1.4.2 Tujuan Khusus

Adapun tujuan khusus dari penelitian ini adalah untuk menerapkan kombinasi algoritma *Rail Fence Cipher* dan *Ceasar Cipher* Untuk Keamanan File.

1.5 Manfaat Penelitian

Adapun manfaat yang akan dikemukakan dari penelitian ini, yaitu sebagai berikut :

1. Untuk menjadi sumber referensi bagi pembaca dalam bidang kriptografi dan kombinasinya khususnya pada algoritma *Rail Fence Cipher* dan *Ceasar Cipher***.**

2. Membantu pengguna dalam mengamankan pesan teks menggunakan kombinasi algoritma *Rail Fence Cipher* dan *Ceasar Cipher***.**

3. Menghasilkan aplikasi yang dapat digunakan untuk mengamankan pesan teks dan penelitian lain mengenai bidang kriptografi

BAB II

LANDASAN TEORI

2.1 Kriptografi

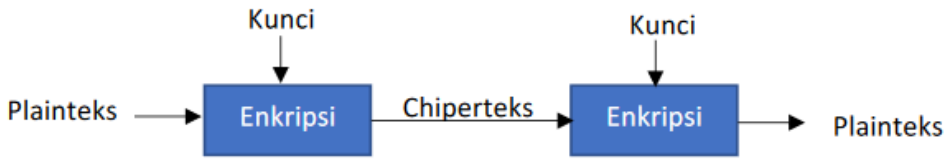
Kriptografi telah dikenal dan dipakai cukup lama sejak kurang lebih tahun 1900 sebelum masehi pada prasasti-prasasti kuburan. Kriptografi sendiri berasal dari kata "*Crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan. Jadi, dapat dikatakan kriptografi adalah tulisan yang tersembunyi. Dengan adanya tulisan yang tersembunyi ini, orang-orang yang tidak mengetahui bagaimana tulisan tersebut disembunyikan tidak akan mengetahui bagaimana cara membaca maupun menerjemahkan tulisan tersebut (Puspita & Wayahdi, 2015).

Criptography berasal dari bahasa Yunani. Menurut bahasanya, istilah tersebut terdiri dari kata kripto dan graphia. Kripto berarti secret (rahasia) dan graphia berarti writing (tulisan) (Fauzah & Iqbal, 2021). Menurut terminologinya, kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan, ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi dari pesan tersebut kemungkinan dapat disadap oleh pihak lain. Untuk menjaga keamanan pesan, maka pesan tersebut dapat discreamble / diacak atau diubah menjadi kode yang tidak dapat dimengerti oleh orang lain (Azlin et al., 2018). Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu himpunan yang berisi elemen plainteks dan himpunan yang berisi elemen chipteks. Enkripsi dan dekripsi merupakan fungsi transformasi antara dua himpunan tersebut.

Enkripsi adalah proses menggunakan algoritma tertentu untuk mengubah data atau informasi menjadi format yang hampir tidak dapat diidentifikasi sebagai informasi asli. *Plaintext* atau teks biasa adalah informasi atau pesan yang dikirim dalam format yang mudah dibaca atau asli (Ziliwu & Maslan, 2022).

Dekripsi adalah kebalikan dari kegiatan enkripsi karena tujuan dari deskripsi mengembalikan pesan yang tersandi atau informasi palsu ke pesan asli. Pada proses mengembalikan isi pesan tersamar harus menggunakan kode yang telah disiapkan sebelumnya. Kegiatan perubahan isi pesan dari plaintext ke ciphertext disebut enkripsi, dan prosedur mengembalikan teks dari ciphertext ke plaintext disebut dekripsi (Ziliwu & Maslan, 2022).

Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi ciphertext, sedangkan dekripsi adalah proses yang dilakukan untuk mengubah pesan tersandi menjadi pesan yang dapat dibaca dan dimengerti (Ridho et al., 2022).



Gambar 2.1 Diagram Enkripsi dan Dekripsi

2.1.1 Tujuan Kriptografi

Kriptografi bertujuan untuk layanan keamanan yang memiliki beberapa aspek keamanan yang memiliki yaitu sebagai berikut (Azlin et al., 2018) :

a. Authentication

Layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi. Fasilitas yang berkaitan untuk melakukan identifikasi terlebih dahulu antara pengirim dan penerima pesan.

b. Integrity

Keuntungan yang didapatkan dalam menggunakan teknik kriptografi yaitu menjamin bahwa pesan akan diterima dalam keadaan masih utuh dan belom mengalamai perubahan selama proses pengiriman. Layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).

c. *Confidelity*

Layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima/pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.

d. Non-repudiation

Layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya.

2.1.2 Jenis Kriptografi

1. Kriptografi`Klasik

Kriptografi klasik merupakan kriptografi yang digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah ditemukan namun belum secanggih sekarang. Kriptografi ini melakukan pengacakan huruf pada kata terang (*plaintext*). Kriptografi ini hanya melakukan pengacakan pada huruf A-Z, dan sangatlah tidak disarankan untuk mengamankan informasi-informasi penting karena dapat dipecahkan dalam waktu singkat. Biarpun telah ditinggalkan, kriptografi klasik tetap dapat ditemui disetiap pelajaran kriptografi sebagai pengantar kriptografi modern (Pardede, Manurung, & Filina, 2017)

2. Kriptografi Modern

Kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Algoritma kriptografi modern (Manaor & Pardede, 2017)

a. Algoritma Simetris

Algoritma simetris adalah algoritma yang menggunakan kunci yang sama untuk enkripsi dan dekripsinya. Algoritma kriprografi simetris sering disebut algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci, dan mengharuskan pengirim dan penerima menyetujui suatu kunci

tertentu. Kelebihan dari algoritma kriprografi simetris adalah waktu proses untuk enkripsi dan dekripsi relatif cepat. Hal ini disebabkan efesiensi yang terjadi pada pembangkit kunci. Karena prosesnya cepat maka algoritma ini tepat untuk digunakan pada sistem komunikasi digital secara *real time.*

b. Algoritma Asimetris

Algoritma Asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan satu lagi *deskripsi*. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia itu, yang dalam hal ini kunci rahasia untuk melakukan pembongkaran terhadap kode yang dikirim untuknya. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA (*Rivest, Shamir dan Adleman).*

2.2 Algoritma Rail Fence Cipher

Rail Fence Cipher merupakan salah satu algoritma cipher transposisi yang mengacak urutan huruf-huruf pesan. Algoritma ini melibatkan penulisan plainteks ke bawah secara berturut-turut yang memiliki baris atas dan baris bawah, sedangkan cipherteksnya diperoleh dengan membaca huruf berdasarkan baris (Girsang, dkk, 2019).

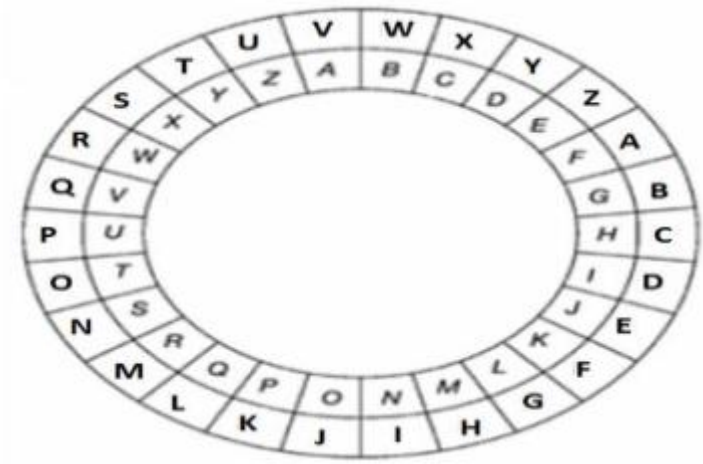
Algoritma *Rail Fence Cipher* menyusun plainteks secara ‘zig-zag’, dengan turun kebawah dan naik keatas sesuai ukuran kolom dan baris yang ditentukan oleh *key*. Cipherteks diperoleh dengan membaca susunan huruf

secara horizontal. *Rail Fence Cipher* pernah digunakan selama perang saudara Amerika, ketika digunakan untuk penyembunyian pesan militer Union maupun mata-mata konfederasi (Kusumaningtyas, 2018)

2.3 Algoritma *Ceasar Cipher*

Caesar cipher dalam ilmu kriptografi adalah metode enkripsi dan deskripsi yang sangat sederhana dan umum. Di dalam caesar cipher tiap huruf disubstitusi dengan huruf berikutnya dari susunan alpabet. Jumlah pergeseran suatu karakter ke karakter lain berdasarkan pada berapa nilai kunci yang dipilih (Ziliwu, Maslan, & Kremer, 2022).

Caesar Cipher menggunakan roda yangberputar dalam enkripsi dan dekripsinya. Roda yang digunakan memiliki dua lingkaran, lingkaran roda yang paling luar dapat diputar bebas (Sutoyo & Murhaban, 2016). Lingkaran roda disajikan seperti Gambar 2.1



Gambar 2.2 Lingkaran Roda Ceasar Cipher

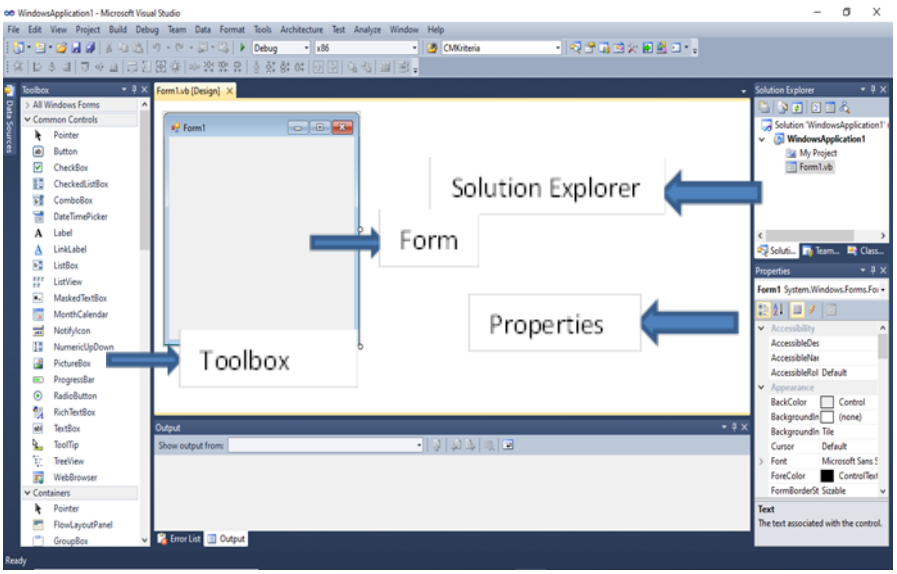
2.4 Tinjauan Bahasa Pemrograman Visual Basic.Net

Visual Studio adalah IDE (*Integrated Development Environment*) yang dapat digunakan untuk mengembangkan aplikasi-aplikasi *Windows*. Visual

studio dirancang untuk fokus pada produktivitas. *Tool* ini disebut juga *Rapid Application DevelopmentTools* (*RAD tools*) karena dirancang dan dilengkapi untuk meningkatkan produktivitas. Versi baru dari Visual Studio inversi terbaru dibuat lebih sederhana untuk mempermudah pengguna dalam mempelajarinya dan memenuhi kebutuhan para *Programme* (Yesputra, 2017).

2.4.1 Menu Utama Integrated Development Environment

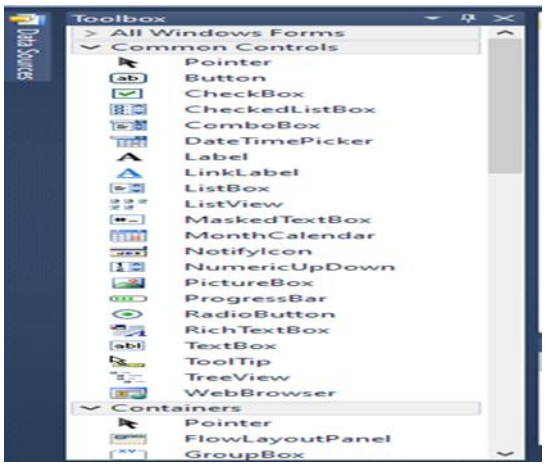
Di dalam menu utama IDE tersedia perintah-perintah dan disertai pula dengan submenu-submenunya. Pada umumnya menu juga dapat ditampilkan dalam bentuk toolbar, tetapi tidak semua opsi tersedia pada saat itu juga. Adakalanya opsi-opsi tersebut tidak dapat diterapkan pada tempat IDE. Ini berarti opsi tersebut dalam keadaan invisible atau disabled.



Gambar 2.3 Komponen Visual Basic 2010

2.4.2 *Toolbox Windows Form*

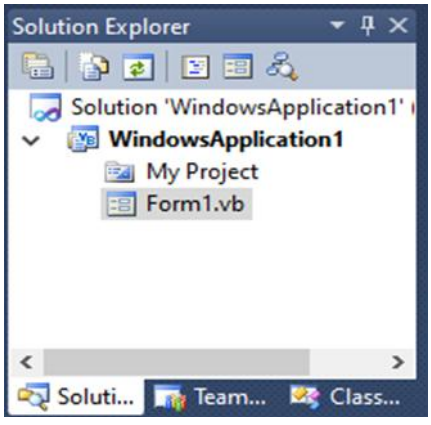
Toolbox berisi berbagai control yang dapat anda gunakan untuk mendesain antarmuka grafis. *Toolbox* mempunyai pengaturan automatic hiding sehingga akan tertutup jika tidak diperlukan.



Gambar 2.4 *Toolbox*

2.4.3 Jendela *Explorer*

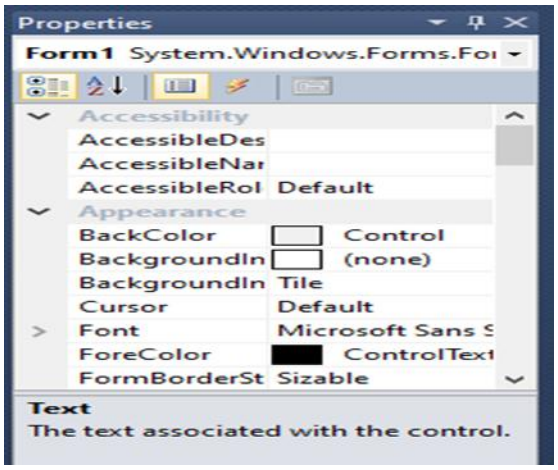
Jendela *explorer* merupakan tempat ditampilkannya daftar-daftar komponen secara hirarki. Dalam Jendela explorer dimungkinkan adanya beberapa proyek, dan dalam proyek ini masih ada beberapa item lagi seperti *form*, *module*, dan lain-lain.



Gambar 2.5 Jendela *Explorer*

2.4.4 Jendela *Properties*

Jendela propertis ini berfungsi untuk menampilkan semua property dari komponen yang dipilih beserta settingannya. Dengan jendela ini kita dapat mengatur property dari masing-masing kontrol yang telah dibuat.



Gambar 2.6 Jendela *Properties*

2.5 Pengertian UML (*Unified Modeling Language)*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. (Suendri, 2018)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau

menggambarkan sebuah sistem software yang terkait dengan objek (Haviluddin, 2016).

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponenkomponen yang diperlukan dalam sistem *software* Adapun tujuan dari UML adalah :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya

Unified Modeling Language(UML) biasa digunakan untuk (Alfina & Harahap, 2019)

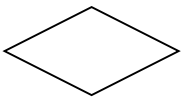
1. Menggambarkan batasan sitem dan fungsi -fungsi sistem secara umum, dibuat dengan *use case* dan *actor*
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction* diagram
3. Menggambarkan representasi struktur statik sebuah sistem dalam betuk *class diagram*

4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan state *transition diagram*
5. Menyatakan arsitektur implementasi fisik menggunakan c*omponent and development*
6. Menyampaikan atau memperluas *fungsionality* dengan *stereo types*

2.6.1 *Activity* Diagram

Activity diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem,yang dimulai dari pandangan business level hingga ke operational level. Pada dasarnya, *activity* diagram merupakan variasi dari statechart diagram. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa. Berikut adalah notasi *activity* diagram

Tabel 2.1 Notasi Activity Diagram

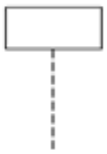
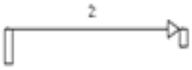
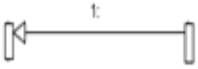
Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk mengambil keputusan

Simbol	Keterangan
	Fork; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

2.6.2 *Sequence* Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek – obyek yang melakukan suatu tugas atau aksi tertentu. Obyek – obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Tabel 2.2. Simbol Sequence Diagram

No.	Gambar	Nama	Keterangan
1		*LifeLine*	objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.6.3 *Use Case* Diagram

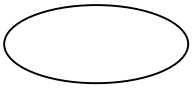
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yangditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana” (*Romi*, 2013). Sebuah *use case*merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuahpekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya.

Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel 2.3. Simbol *Use Case* Diagram

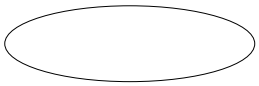
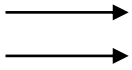
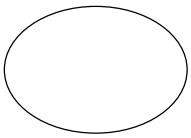
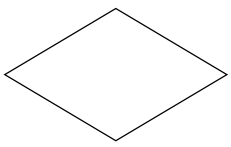
Gambar	Nama	Keterangan
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinterasi dengan *Use Case*.
	*Depedency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	*Generalization*	Hubungan dimana objek anak(*Descended)* berbagi prilaku dan struktur data dari objek yang diatasnya objek induk.
	*Include*	Menspesifikasikan bahwa use case sumber secara explicit.
	*Extend*	Menspesifikasikan bahwa use case target memperluas prilaku pada use case sumber pada sebuah titik diberikan.

Gambar	Nama	Keterangan
	Assosiation	Apa yang memhubungkan objek satu dengan objek yang lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur dari sebuah *actor*.
	*Colaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya.
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.7 Flowchart

Flowchart dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus. Manfaat yang akan diperoleh bila menggunakan flowchart dalam pemecahan masalah komputasi: Terbiasa berfikir secara sistematis dan terstruktur, mudah mengecek dan menemukan bagian-bagian prosedur yang tidak valid dan bertele-tele. Prosedur akan mudah dikembangkan (Nuraini, 2015).

Tabel 2.4 Simbol dan Fungsi *Flowchart*

SIMBOL	KETERANGAN
	Star/Mulai End/ Selesai
	Simbol arus/ flow yang menyatakan jalannya proses
	Simbol connector, (menyatakan sambungan dari proses ke proses lainnya dalam hal yang sama
	Simbol process yaitu menyatakan suatu tindakan
	Simbol manual, manyatakan suatu tindakan
	Simbol *decision*, menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan

SIMBOL	KETERANGAN
	Simbol *keying* operation menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
	Simbol *input*/*output* menyatakan proses input/output
	Simbol dokumen mencetak keluaran dalam bentuk dokumen

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu penelitian dimulai dari 01 September 2022 sampai dengan 31 Mei 2023. Penelitian dilakukan secara mandiri dengan melakukan ujicoba terhadap file dan menganalisa dari penerepan algoritma yang digunakan.

3.2 Metode Penelitian

Dalam melakukan penelitian ini, penulis menggunakan metode terapan (*applied research*). Penelitian terapan atau *applied research* dilakukan berkenaan dengan kenyataan-kenyataan praktis, penerapan, dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Penelitian terapan berfungsi untuk mencari solusi tentang masalah-masalah tertentu. Tujuan utama penelitian terapan adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk kepentingan manusia baik secara individu atau kelompok maupun untuk keperluan industri atau politik dan bukan untuk wawasan keilmuan semata.

Dalam melaksanakan penelitian terapan ini terdapat 5(lima) langkah, diantaranya :

a. Melakukan sesuatu yang sedang diperlukan, dipelajari, diukur, dan diperiksa kelemahannya.
b. Mencari satu dari kelemahan-kelemahan yang diperoleh dipilih untuk penelitian.

c. Mencari dan memberikan solusi dalam melakukan pemecahan masalah
d. Kemudian dilakukan modifikasi sehingga penyelesaian dapat dilakukan untuk diterapkan.
e. Pemecahan dipertahankan dan menempatkannya dalam suatu kesatuan sehingga jadi bagian permanen dalam satu sistem.

3.3 Perangkat Keras dan Perangkat Lunak

Adapun perangkat keras dan perangkat lunak yang digunakan dalam mengidentifikasi kata pada citra digital dapat dilihat di tabel 3.1.

Tabel 3.1 Perangkat Keras dan Perangkat Lunak

Perangkat Keras	**Perangkat Lunak**
1. Laptop Lenovo 2. Harddisk 500 Gb 3. RAM 4GB 4. Mouse Logitec 5. Printer.	1. Sistem Operasi Windows 10 2. Bahasa Pemrograman Visual Basic.Net 2010.

3.4 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh data yang dapat mendukung permasalahan yang akan dibahas. Sehubungan dengan hal ini maka digunakan metode pengumpulan data yang meliputi :

a. Observasi

Dalam pengumpulan data melalui observasi, penulis mengamati dan menganalisa bagaimana cara sistem mampu melakukan enkripsi dan dekripsi *file*.

b. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan yang berupa karya ilmiah, jurnal, buku-buku serta dari *internet* yang berhubungan dengan penulisan ini. Tujuan dari studi pustaka ini adalah untuk mendalami dan memperoleh keterangan yang lengkap terhadap objek yang diteliti.

3.5 Analisa Perancangan Sistem

3.5.1 Analisa Kebutuhan Fungsional Sistem

Kebutuhan fungsional yang harus dimiliki oleh sistem kriptografi *Rail Fence Cipher* dan *Ceasar Cipher* adalah :

1. Sistem dapat mengenkripsi pesan dengan baik
2. Sistem dapat mendekripsi ke pesan asli dengan baik

3.5.2 Analisis Kebutuhan *Non*-Fungsional Sistem

Kebutuhan *non*-fungsional yang harus dimiliki oleh sistem pengamanan *file* menggunakan metode *Rail Fence Cipher* dan *Ceasar Cipher* adalah :

1. Sistem memiliki proses yang akurat dan cepat
2. Tampilan antarmuka sistem menarik dan dapat dimengerti oleh pengguna sistem

3.5.3 Analisa Sistem Aktual

Masalah yang diangkat dari penelitian ini adalah pembuatan sistem pengamanan *file* teks menggunakan kombinasi algoritma *Rail Fence Cipher* dan *Ceasar Cipher*. Dimana algoritma *Rail Fence Cipher* dan *Ceasar Cipher* merupakan metode yang dapat digunakan untuk mengenkripsi *file* teks yang berekstensi txt dan rtf. Metode *Rail Fence Cipher* dan *Ceasar Cipher* merupakan kripto sistem yang menggunakan algoritma simetris.

3.5.4 Analisa Algoritma *Rail Fence Cipher* dan *Ceasar Cipher*

A. Proses Enkripsi *Rail Fence Cipher*

Proses enkripsi pada algoritma *Rail Fence Cipher* menggunakan kunci yang diberikan oleh pengguna. Adapun proses enkripsi pada algoritma *Rail Fence Cipher* adalah sebagai berikut :

Diketahui plainteks yang akan di enkripsi :

"UNIVERSITAS DEHASEN"

Kunci Enkripsi = "FILKOM"

Berikut proses enkripsi menggunakan metode *Rail Fence Cipher* :

1. Mengurutkan kunci berdasarkan urutan abjad.

   "F" = 1

   "I" = 2

   "L" = 4

   "K" = 3

   "O" = 6

   "M" = 5

2. Membentuk matriks transposisi berukuran m x n yang mana lebar matriks diperoleh dari panjang kunci.

F	I	L	K	O	M
1	2	4	3	6	5

3. Menyusun karakter plainteks ke dalam matriks.

F	I	L	K	O	M
1	2	4	3	6	5
U	N	I	V	E	R
S	I	T	A	S	D
E	H	A	S	E	N

4. Membaca chiperteks dengan cara membaca secara kolom urutan karakter plainteks pada matriks transposisi mulai dari kode kunci yang paling kecil.

   Membaca kolom kunci “F” :

   USE

   Membaca kolom kunci “I” :

   NIH

   Membaca kolom kunci “K”

   VAS

   Membaca kolom kunci “L”

   ITA

   Membaca kolom kunci “M”

   RDN

   Membaca kolom kunci “O”

   ESE

5. Menggabungkan pembacaan pada langkah 6 sehingga menghasilkan chiperteks :

   USENIHVASITARDNESE

3.5.5 Proses Enkripsi *Ceasar Cipher*

Proses enkripsi pada algoritma *Ceasar Cipher* mengunakan chiperteks dari *Rail Fence Cipher* sebagai plainteks. Adapun proses enkripsi pada algoritma *Ceasar Cipher* adalah sebagai berikut Diketahui plainteks yang akan di enkripsi :

"USENIHVASITARDNESE"

Plainteks : USENIHVASITARDNESE

Key : FILKOM

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	I	L	K	O	M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T

Hasil Enkripsi Ceaser Cipher : OMOHCBPFMCNFLKHOMO

Pesan	U	S	E	N	I	H	V	A	S	I	T	A	R	D	N	E	S	E
Enkripsi	O	M	O	H	C	B	P	F	M	C	N	F	L	K	H	O	M	O

Hasil Dekripsi Ceasar Cipher : USENIHVASITARDNESE

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	I	L	K	O	M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T

Enkripsi	O	M	O	H	C	B	P	F	M	C	N	F	L	K	H	O	M	O
Dekripsi	U	S	E	N	I	H	V	A	S	I	T	A	R	D	N	E	S	E

Setelah diperoleh hasil dari dekripsi algoritma Ceaser Cipher, selanjut melakukan dekripsi dengan menggunakan algoritma Rail Fence Cipher, adapun prosesnya adala sebagai berikut :

1. Menyusun karakter chiperteks mulai dari indeks kunci paling kecil

USENIHVASITARDNESE

Kolom ”F” = USE

F	I	L	K	O	M
1	2	4	3	6	5
U	N				
S					
E					

Kolom “I” = NIH

F	I	L	K	O	M
1	2	4	3	6	5
U	N				
S	I				
E	H				

Kolom “K” = VAS

F	I	L	K	O	M
1	2	4	3	6	5
U	N	I	V		
S	I		A		
E	H		S		

Kolom “L” = ITA

F	I	L	K	O	M
1	2	4	3	6	5
U	N	I	V		
S	I	T	A		

E	H	A	S		

Kolom “M” = RDN

F	I	L	K	O	M
1	2	4	3	6	5
U	N	I	V		R
S	I	T	A		D
E	H	A	S		N

Kolom “O” = ESE

F	I	L	K	O	M
1	2	4	3	6	5
U	N	I	V	E	R
S	I	T	A	S	D
E	H	A	S	E	N

2. Membaca karakter secara baris dari matriks.

   Baris 1 : UNIVER

   Baris 2 : SITASD

   Baris 3 : EHASEN

3. Menggabungkan karakter sehingga diperoleh plainteks

   “UNIVERSITASDEHASEN”

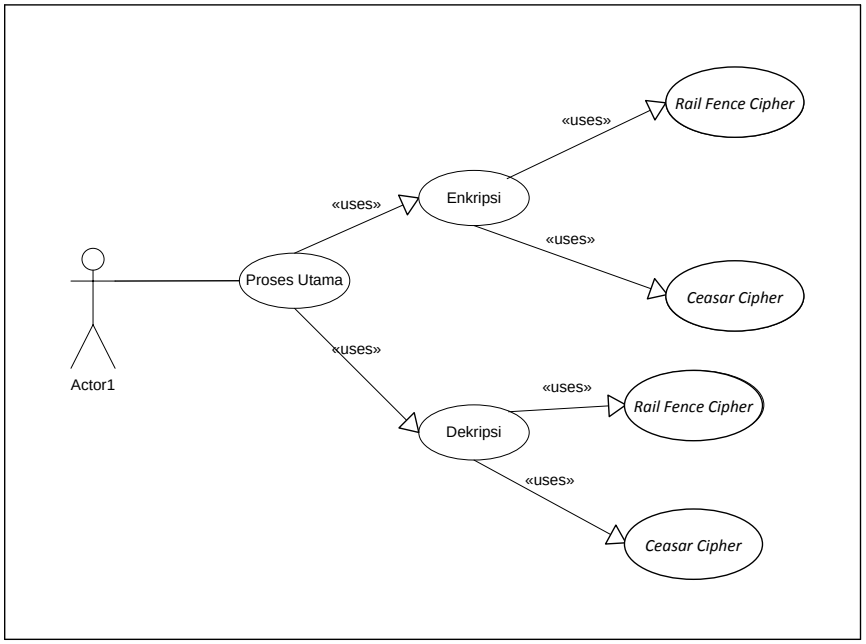
3.5.6 Analisa Perancangan Sistem Baru

Perancangan sistem yang akan dilakukan pada penelitian ini terbagi menjadi dua tahap, yaitu perancangan *flowchart* dan perancangan antarmuka. Perancangan

perancangan *flowchart diagram* bertujuan untuk memberikan gambaran mengenai proses dan alur penggunaan dari sistem yang dikembangkan. Sedangkan perancangan antarmuka dilakukan untuk memberikan gambaran tampilan antar muka dari sistem yang dikembangkan.Berikut penjabaran dari masing – masing perancangan yang dilakukan pada penelitian ini :

1. Perancangan Use Case

Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara 'aktor'—inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana. Berikut adalah use case yang digunakan pada sistem yang dirancang oleh penulis



Gambar 3.1 Use Case Sistem

Gambar 3.3 menyatakan diagram use case sistem kriptografi untuk keamanan file. Use Case ini menjelaskan mengenai bagaimana proses pengenkripsian dan dekripsi menggunakan kedua algoritma *Rail Fence Cipher* dan *Ceasar cipher*, untuk menyandikan file yang ingin di jaga kerahasiaannya

Tabel 3.2 Spesifikasi *Use Case Input Teks*

Nama Use Case	*Input Teks*	
Aktor	*User*	
Deskripsi	*Use Case* ini mendeskripsikan bagaimana proses menginput teks baik plainteks pada proses enkripsi maupun chiperteks pada proses dekripsi	
Kegiatan *User*		Respon Sistem
Alur Dasar	1. Memilih tombol ”Open”	1. Menampilkan dialog penyimpanan file untuk memilih file teks yang akan digunakan
	2. Tekan tombol *”Ok”*	2. Melakukan pembacaan karakter dari pesan yang dipilih dan menyimpannya ke dalam variabel untuk di proses lebih lanjut.
Alur Alternatif	Tidak ada	
Kondisi Sesudah	Sistem menampung karakter pesan teks yang dipilih pengguna	

Tabel 3.3 Spesifikasi *Use Case* Enkripsi

Nama Use Case	*Enkripsi*	
Aktor	*User*	
Deskripsi	*Use Case* ini mendeskripsikan bagaimana proses enkripsi	
Kegiatan *User*		Respon Sistem

Alur Dasar	1. Mengisi kunci	1. Membaca kunci
	2. Mengisi plainteks	2. Membaca plainteks dan mengkonversikannya ke deretan karakter
	3. Tekan tombol proses	3. Melakukan enkripsi deretan karakter pada pesan menggunakan *Rail Fence Cipher*dan *Ceasar Cipher*
Alur Alternatif	Tidak ada	
Kondisi Sesudah	Sistem menghasilkan chiperteks hasil enkripsi	

Tabel 3.4 Spesifikasi *Use Case* Dekripsi

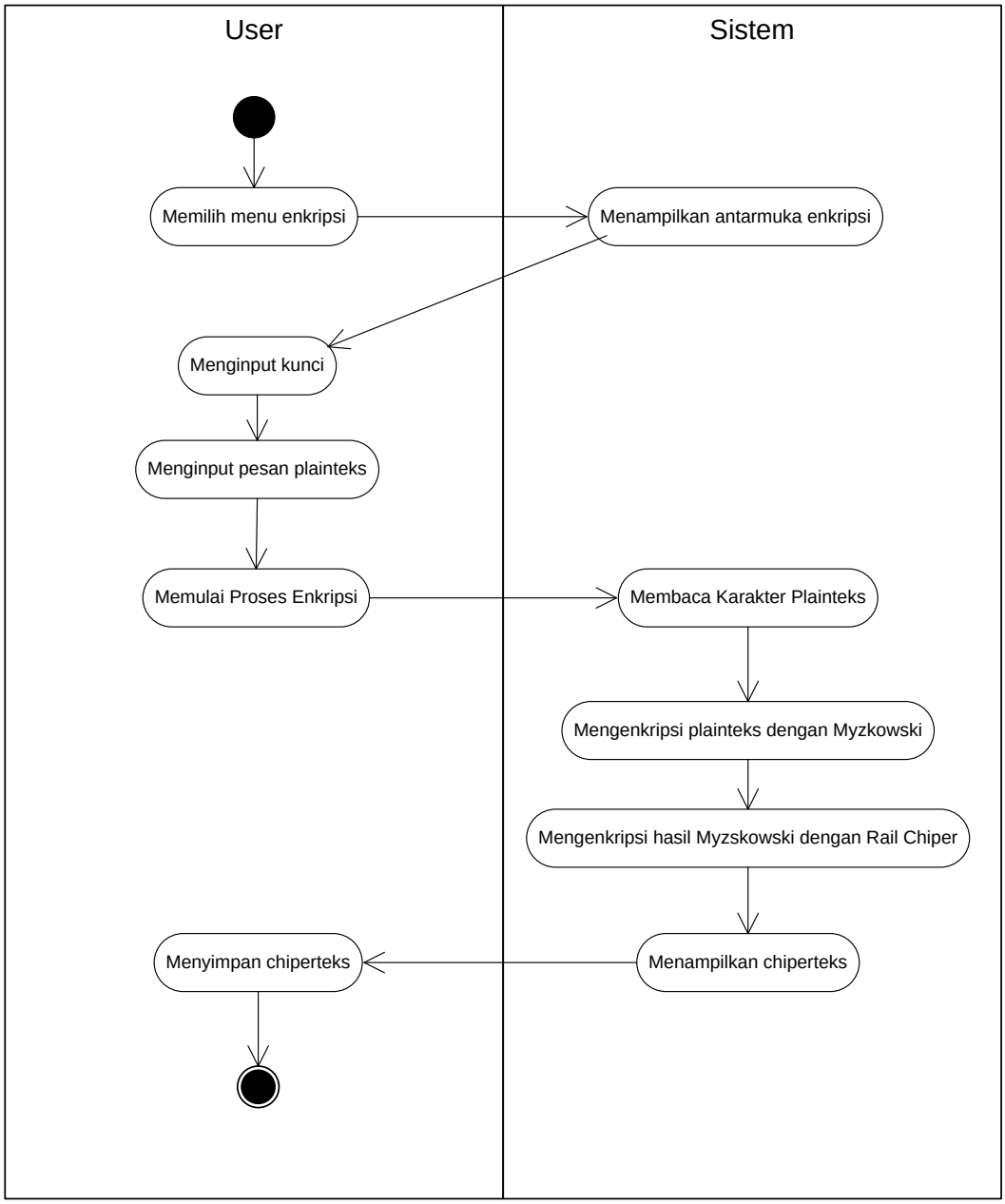
Nama Use Case	*Dekripsi*	
Aktor	*User*	
Deskripsi	*Use Case* ini mendeskripsikan bagaimana proses dekripsi	
Kegiatan *User*		Respon Sistem
Alur Dasar	1. Mengisi kunci	1. Membaca kunci
	2. Mengisi chiperteks	2. Membaca chiperteks dan membagi chiperteks menjadi deretan karakter.
	3. Tekan tombol Proses	3. Melakukan dekripsi nilai chiperteks menggunakan *Rail Fence Cipher*dan *Vigenere Chiper*
Alur Alternatif	Tidak ada	
Kondisi Sesudah	Sistem menghasilkan plainteks	

Tabel 3.5 Spesifikasi *Use Case* Simpan Teks

Nama Use Case	*Simpan Teks*	
Aktor	*User*	
Deskripsi	*Use Case* ini mendeskripsikan bagaimana proses penyimpanan teks baik dari hasil enkripsi maupun hasil dari dekripsi	
Kegiatan *User*		Respon Sistem
Alur Dasar	1. Menekan tombol ”Open”	1. Menampilkan dialog untuk memilih lokasi dan nama file.
	2. Menekan tombol ”Save”	2. Menyimpan karakter pesan hasil enkripsi ataupun dekripsi kedalam file di media penyimpanan.
Alur Alternatif	Tidak ada	
Kondisi Sesudah	Sistem menghasilkan file baru pada media penyimpanan	

***3. ActivityDiagram*Enkripsi**

Tampilan *Activity Diagram* Enkripsi sebagai berikut:



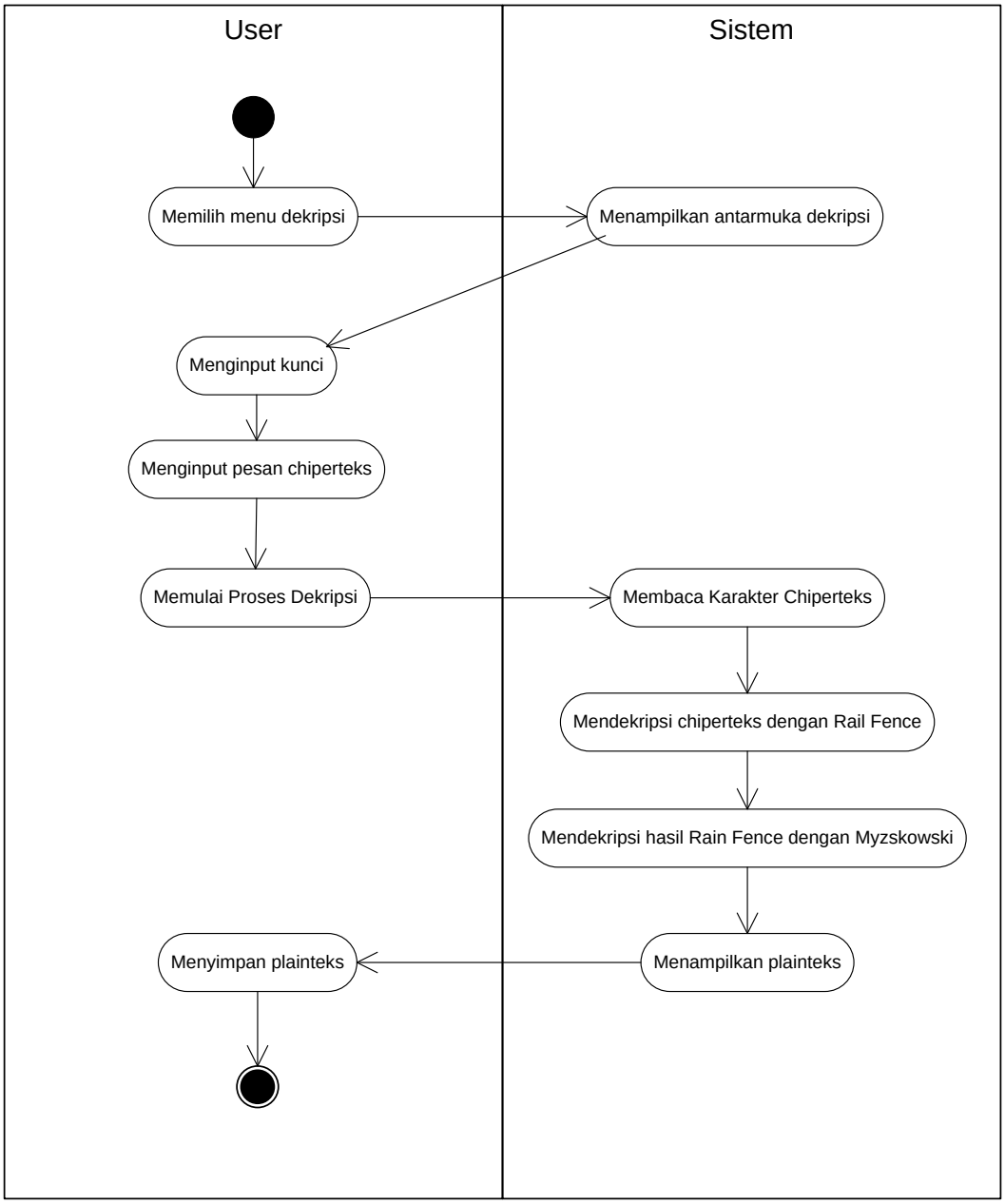
Gambar 3.5 *DiagramActivity* Enkripsi

Activity Diagram enkripsi seperti yang terlihat pada gambar diatas menunjukkan tahapan aktivitas pengguna dalam berinteraksi dengan sistem dimana pengguna menginputkan kunci dan pesan plainteks yang akan dienkripsi. Sistem kemudian melakukan pembacaan karakter pesan dan melakukan enkripsi pesan plainteks menggunakan *Rail Fence Cipher*dan *Vigenere Chiper*. Hasil enkripsi berupa

chiperteks ditampilkan kepada pengguna yang kemudian dapat disimpan untuk digunakan lebih lanjut.

3. ***Diagram Activity* Dekripsi**

Tampilan *Activity Diagram* Dekripsi sebagai berikut:



Gambar 3.2.*Diagram Activity* Dekripsi

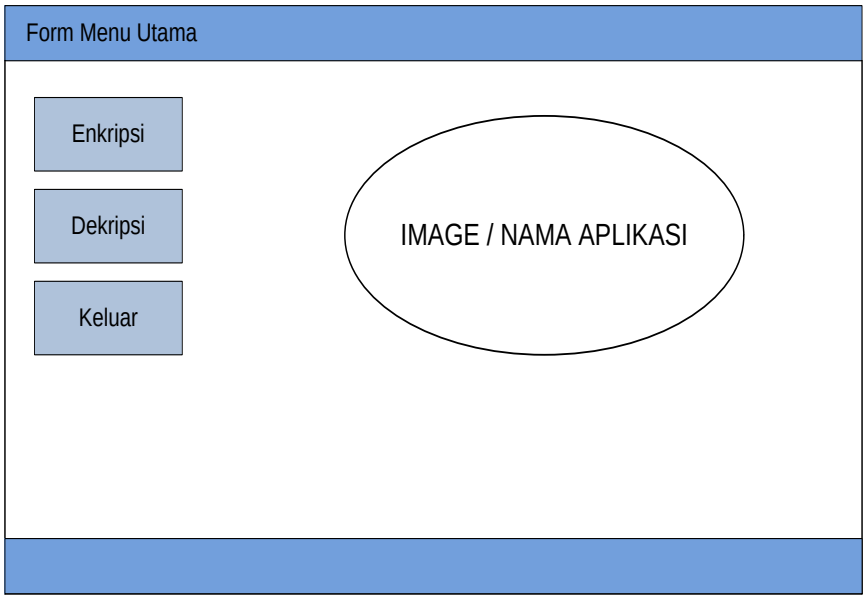
Diagram Activity dekripsi seperti yang terlihat pada gambar dibawah menunjukkan tahapan aktivitas pengguna dalam berinteraksi dengan sistem dimana pengguna menginputkan kunci dan pesan chiperteks yang akan didekripsi. Sistem kemudian melakukan pembacaan karakter chiperteks dan melakukan dekripsi pesan chiperteks menggunakan urutan *Ceasar Cipher* dan *Myzskowski Transposition*. Hasil dekripsi berupa plainteks ditampilkan kepada pengguna yang kemudian dapat disimpan untuk digunakan lebih lanjut

3.5.7 Rancangan Antarmuka

Perancangan ini bertujuan untuk merancang tampilan dari suatu perangkat lunak yang akan di buat yang sesuai dengan kebutuhan pengguna. Berikut perancangan antarmukaaplikasi enkripsi dan dekripsi pesan menggunakan algoritma *Vigenere* dan *Ceaser Cipher*

A. Menu Utama

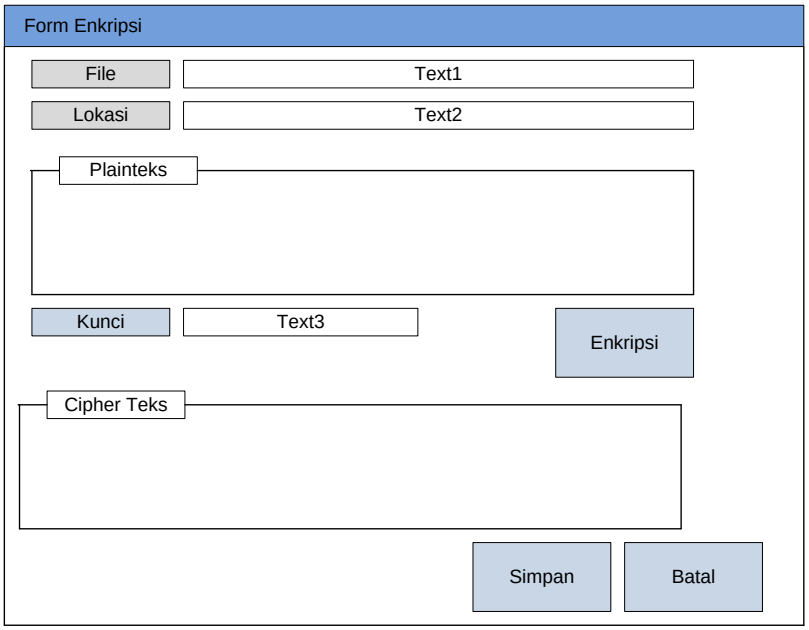
Halaman utama merupakan halaman yang tampil pada saat aplikasi pertama dijalankan. Halaman ini berisi logo, pesan selamat datang dan menu utama.



Gambar 3.3 Rancangan Menu Utama Aplikasi

B. Rancangan Antarmuka Enkripsi

Halaman ini merupakan halaman enkripsi dimana pengguna dapat melakukan enkripsi pesan menggunakan kunci yang diinginkan. Berikut rancangan halaman enkripsi



Gambar 3.4 Rancangan Form Enkripsi

Keterangan dari gambar 3.4 adalah:

1. *File*

   Komponen ini merupakan komponen tombol yang digunakan untuk membuka file teks yang akan dienkripsi

2. *Plainteks*

   Komponen ini merupakan komponen menerima inputan plainteks .

3. *Chiperteks*

   Komponen ini merupakan komponen untuk menampung hasil enkripsi.

4. Enkripsi

   Komponen ini merupakan komponen tombol yang digunakan untuk mengenkripsi.

5. Simpan

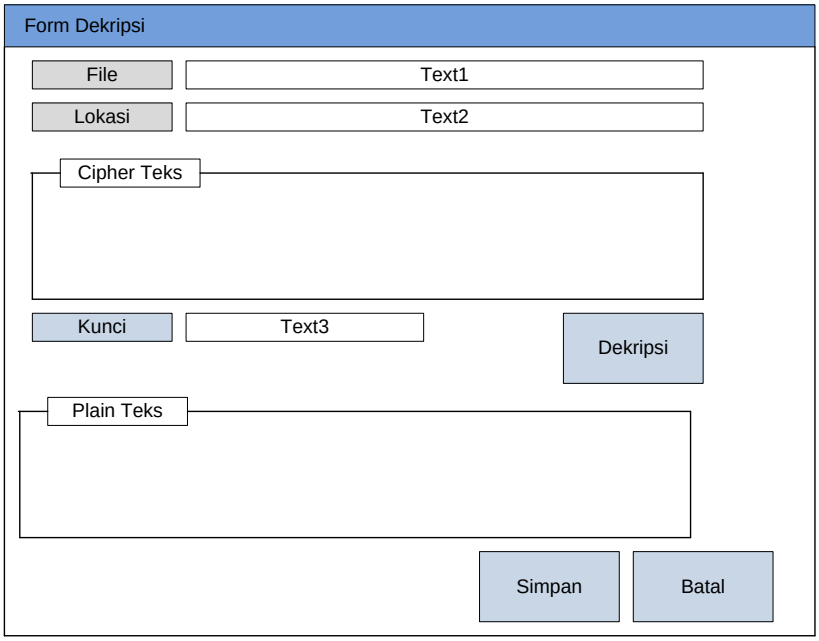
   Komponen ini merupakan komponen tombol yang digunakan untuk menyimpan hasil enkripsi

6. Batal

   Komponen ini merupakan komponen tombol yang digunakan untuk membatalkan proses enkripsi.

C. Rancangan Antarmuka Dekripsi

Halaman ini merupakan halaman dekripsi dimana pengguna dapat melakukan dekripsi pesan *chiperteks* menggunakan kunci yang diinginkan. Berikut rancangan halaman dekripsi



Gambar 3.5 Rancangan Form Dekripsi

Keterangan dari gambar 3.5 adalah:

1. *File*

   Komponen ini merupakan komponen tombol yang digunakan untuk membuka file teks yang akan didekripsi

2. *Chiperteks*

   Komponen ini merupakan komponen untuk menerima inputan *chipertext*.

3. *Plainteks*

   Komponen ini merupakan komponen menampung hasil dekripsi.

4. Dekripsi

   Komponen ini merupakan komponen tombol yang digunakan untuk mendekripsi.

5. Simpan

Komponen ini merupakan komponen tombol yang digunakan untuk menyimpan hasil dekripsi.

6. Batal

   Komponen ini merupakan komponen tombol yang digunakan untuk membatalkan proses dekripsi

3.6 Perancangan Pengujian

Pengujian sistem merupakan proses eksekusi sistem dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau output yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Metode pengujian yang dipakai dalam sistem ini adalah metode *black box.* Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program. Tujuan metode ini mencari kesalahan pada:

1. Fungsi yang salah atau hilang.
2. Kesalahan pada *interface*.
3. Kesalahan pada struktur data atau akses database.
4. Kesalahan performansi.
5. Kesalahan inisialisasi dan tujuan akhir