# **PROPOSAL**

**IMPLEMENTASI ALGORITMA *LEVENSHTEIN DISTANCE* PADA SISTEM INFORMASI PENELUSURAN TUGAS AKHIR BERBASIS *WEB***

**(Studi Kasus: Jurusan Teknik Informatika Universitas Halu Oleo)**

**Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik**

****

**MUH. DARUL ZULKIFLI RITOM**

**E1E115076**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**KENDARI**

**2021**

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Perkembangan teknologi informasi saat ini begitu pesat, membuat banyak orang bergantung pada kemudahan yang ditawarkan teknologi informasi, salah satunya komunikasi yang dapat dilakukan melaui telepon maupun pesan teks. Kegiatan mengirim pesan telah menjadi budaya massa pada zaman modern ini, pesan teks sendiri dapat digunakan untuk berinteraksi secara sistematik. Salah satu fitur yang paling banyak digunakan adalah SMS (Editor, 2012). Dengan perkembangan teknologi, aplikasi pesan yang dulunya menggunakan SMS kini mulai berganti menjadi aplikasi yang dapat diakses menggunakan internet. Seiring dengan berjalannya waktu, muncul berbagai macam masalah yang kerap dijumpai dalam aplikasi berkirim surat ini terutama dalam hal keamanan informasi (Sugiantoro, 2012).

Keamanan informasi merupakan hal yang perlu diperhatikan saat bertukar pesan melalui dunia maya, terutama jika pesan yang ingin disampaikan tersebut bersifat penting dan rahasia dan hanya boleh diketahui oleh pihak-pihak tertentu. Pengiriman pesan atau informasi tanpa dilakukan pengamanan akan memiliki resiko terhadap penyadapan dan informasi yang dikirim dapat dengan mudah diketahui oleh pihak-pihak yang tidak berhak. Oleh karena itu perlu digunakan berbagai cara untuk mengamankan pesan tersebut agar dapat sampai ke tujuan secara utuh. Pengamanan pesan dapat dilakukan dengan berbagai cara, antara lain kriptografi dan steganografi.

Umumnya untuk menjaga kerahasiaan dan keamanan informasi ini dilakukan dengan enkripsi. Enkripsi adalah proses mengamankan informasi dengan membuatnya tidak dapat dibaca tanpa adanya pengetahuan khusus tentang enkripsi tersebut. Enkripsi menggunakan suatu algoritma yang dapat mengkodekan semua aliran data bit dari sebuah pesan menjadi *cryptogram* yang tidak dapat dimengerti dan untuk memperoleh kembali informasi yang asli dilakukan proses deskripsi dengan menggunakan kunci yang benar.

Namun, informasi yang dienkripsi tidak jarang menimbulkan rasa curiga bagi sekelompok orang dan keinginan untuk memecahkan informasi tersebut. Penelitian yang dilakukan oleh Kiltz dan Piertzak (2010) menjelaskan bahwa banyak terjadi serangan terhadap data user, meskipun data tersebut telah dienkripsi. Enkripsi tidak dapat menjamin keamanan data secara utuh, bahkan banyak data yang telah dienkripsi dapat dipecahkan. Banyaknya teknik kriptoanalisis yang telah berkembang saat ini menyebabkan kemungkinan untuk memecahkan sebuah enkripsi lebih besar.

Lebih lanjut, hasil riset menunjukkan bahwa semakin maraknya penggunaan sistem enkripsi untuk mengamankan data justru menciptakan kondisi yang sempurna bagi para penjahat *cyber* untuk menyelipkan *malware* di dalam data transaksi yang dienkripsi, dan bahkan mengurangi kerumitan pembuatan *malware* cerdas yang bisa menghindari deteksi keamanan (Heriyanto, 2014).

Oleh karena itu, perlu ditambahkan sebuah sistem pendukung pada pengamanan data setelah melakukan teknik kriptografi yaitu dengan teknik penyembunyian data atau disebut steganografi.

Kriptografi merupakan seni dan ilmu untuk memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditujukan untuk orang yang hanya memiliki sebuuah kunci untuk mengubah kode itu kembali yang berfungsi dalam menjaga kerahasiaan data atau pesan.

RC6 adalah salah satu algoritma dalam kriptografi yang merupakan algoritma cipher blok baru yang didaftarkan ke NIST yang diajukan oleh RSA *Security Laboratories*. Algoritma ini dirancang oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin untuk mengikuti kontes *Advanced Ecryption Standard* (AES) dan berhasil menjadi salah satu dari lima finalisnya. Rancangan berawal dari keinginan untuk meningkatkan tingkat keamanan dari RC5 untuk dapat memenuhi standar dari kontes tersebut (Sujadi dkk., 2015).

RC6 adalah algoritma yang memiliki struktur yang sederhana dan cepat sehingga mudah diaplikasikan untuk pengamanan data, biasanya dipakai untuk mengacak *plain text*. Tingkat keamanan pada algoritma ini terletak pada kekuatan rotasi yang berdasarkan data, penggunaan eksklusif OR yang bergantian, fungsi modulo dan fungsi persamaan yang menggunakan rotasi yang tetap (Juliansyah, 2017).

Steganografi dapat dikatakan mempunyai hubungan yang erat dengan kriptografi, tapi metode ini sangat berbeda dengan kriptografi. Kriptografi mengacak pesan sehingga tidak dimengerti, sedangkan steganografi menyembunyikan pesan sehingga tidak terlihat. Pesan dalam *ciphertext* mungkin akan menimbulkan kecurigaan sedangkan pesan yang dibuat dengan steganografi tidak akan menimbulkan kecurigaan. Kedua teknik ini dapat digabungkan untuk mendapatkan metode pengiriman rahasia yang sulit dilacak. Metode yang banyak digunakan dalam steganografi adalah *Least Significant Bit* (LSB) yaitu metode yang memilih secara acak piksel yang ingin disisipkan oleh pesan rahasia (Junaidi, 2015).

Pada tahun 2015, Win Junaidi, dalam penelitiannya yang berjudul “*Algoritma Hill Chiper Untuk Enkripsi Data Teks Yang Digunakan Untuk Steganografi Gambar Dengan Metode LSB (Least Significant Bit)”*, membahas pengamanan pesan teks dengan cara menggabungkan dua model pengamanan yaitu kriptografi dengan metode *Hill Cipher* dan steganografi dengan metode LSB, yang mempunyai tahapan pesan teks akan dienkripsi terlebih dahulu untuk membuat pesan teks tersebut tidak bisa dibaca kemudian hasil enkripsi tersebut dimasukkan ke gambar setelah itu dilakukan proses dekripsi agar pesan tersebut bisa terbaca kembali.

Metode LSB menggunakan citra digital sebagai *covertext*. Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), ada bit yang paling berarti (*most significant bit* atau MSB) dan bit yang paling kurang berarti (*least significant bit* atau LSB) (Rahmansyah, 2019). Metode ini menyembunyikan pesan dengan mengganti bit-bit data yang kurang berarti dalam segemen citra dengan bit-bit rahasia pada segmen terakhir. Dengan mengganti bit-bit yang kurang berarti perubahan yang terjadi hanya mengubah nilai byte satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya, dimana mata manusia tidak dapat membedakan perubahan kecil tersebut.

Berdasarkan latar belakang tersebut, dalam penelitian ini akan dibuat Implementasi RC6 Untuk Penyembunyian Pesan Teks Pada Steganografi Gambar Menggunakan Metode LSB untuk keamanan pesan teks yang lebih baik.

* 1. **Rumusan Masalah**

Berdasarkan uraian pada latar belakang masalah, maka yang menjadi perumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan algoritma RC6 sebagai salah satu metode mengamankan data dengan menggunakan metode LSB.

* 1. **Batasan Masalah**

Adapun batasan masalah dalam penelitian ini yaitu:

1. Gambar yang menampung pesan teks yang telah terenkripsi pada penelitian adalah gambar yang berekstensi .png.
2. Panjang maksimum teks yang dimasukkan tergantung jumlah pixel yang ada pada gambar.
3. Sistem yang akan dirancang menggunakan bahasa pemrograman PHP 5, dan HTML 5.
   1. **Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma RC6 sebagai salah satu metode mengamankan data dengan menyisipkan pesan teks yang terenkripsi pada gambar menggunakan metode LSB.

* 1. **Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah dapat memahami bagaimana proses RC6 untuk diimplementasikan pada gambar dengan menggunakan metode LSB.

* 1. **Sistematika Penulisan**

**BAB 1 PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

**BAB II LANDASAN TEORI**

Pada bab ini, membahas teori mengenai keaman data, kriptografi, RC6, steganografi, dan LSB.

**BAB III METODOLOGI PENELITIAN**

Bab ini memuat metodologi penelitian yang meliputi metode pengumpulan data, uraian metode *Rational Unified Process* (RUP) untuk pengembangan sistem dan uraian waktu penelitian.

**BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Bab ini memuat analisis serta rancangan sistem yang akan dibuat. Adapaun rancangan sistem meliputi rancangan proses, rancangan UML dan rancangan antarmuka sistem.

**BAB V IMPLEMENTASI DAN PENGUJIAN**

Bab ini menyajikan implementasi dan pengujian dari sistem yang dibangun. Bagian implementasi menguraikan tentang implementasi secara detail dan runut dari sistem yang dibangun berdasarkan hasil analisis dan rancangan pada bab sebelumnya sedangkan bagian pengujian menguraikan pengujian sistem serta pembahasan hasil pengujian sistem tersebut.

**BAB VI PENUTUP**

Bab ini memuat kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian selanjutnya.

* 1. **Tinjauan Pustaka**

Beberapa penelitian yang dijadikan rujuan dalam penelitian ini yaitu : Sandro Sembiring (2013) yang melakukan penelitian tentang perancangan aplikasi steganografi untuk menyisipkan pesan teks pada gambar dengan metode *End of File.* Pada penelitian ini langkah langkah atau cara menyisipkan pesan atau proses encoding pesan adalah: pertama pesan diubah ke bilangan biner, Kemudian dapatkan nilai *pixel* gambar, Setelah itu urutkan nilai *pixel* gambar sesuai urutan dari yang terkecil, kemudian melakukan iterasi variabel untuk menentukan letak pesan yang akan disisipkan dan melakukan proses penyisipan pesan kedalam berkas RGB.Selanjutnya langkah langkah atau proses pengambilan pesan atau proses *Decoding* pesan adalah :cPertama tentukan nilai m dari proses *encoding* kemudian tentukan posisi dari nilai *integer* warna gambar tersebut, kemudian beri nilai m=0 dan selanjutnya proses iterasi terhadap *variable* nilai palet warna tersebut agar pesan asli atau *plantext* didapat kembali. Kemudian Menerapkan metode *End of File* adalah dengan melakukan proses *encoding* dengan langkah pertama rubah pesan ke biner, kemudian dapatkan nilai *pixel* gambar ,kemudian urutkan palet warna, kemudian lakukan iterasi untuk menetukan letak pesan yang disisipkan pada berkas RGB tersebut. Selanjutnya proses *decoding* pesan dengan langkah dapatkan nilai m pesan agar pesan mudah didapat, kemudian melakukan proses iterasi pesan agar pesan yang telah disisipkan didapat kembali.

Penelitian yang kedua dilakukan oleh Win Junaidi (2015) dengan judul algoritma *Hill Chiper* untuk enkripsi data teks yang digunakan untuk steganografi gambar dengan metode LSB (*Least Significant Bit*). Pada penelitian ini cara untuk mengenkripsi teks dengan algoritma *hill chiper* adalah dengan mengubah masing masing huruf teks menjadi matriks angka dengan memanfaatkan tabel kode yang telah ditentukan kemudian hasil perkalian antara matriks pesan dan kunci yang telah di modulus dengan 26 (duapuluh enam) diubah kembali menjadi urutan huruf dengan tabel kode. Untuk proses dekripsi sama seperti proses enkripsi tetapi bedanya adalah matriks kunci di invers terlebih dahulu baru kemudia dilakukan perkalian dan modulus 26 hasil perkalian akan mengembalikan chiper ke bentuk pesan semula dan untuk menerapkan chiperteks yang telah diekripsi dengan algoritma *hill chiper* ke dalam proses steganografi sama dengan menerapkan pesan biasa yang belum di sandikan atau dienkripsi, yaitu dengan mengubah pesan menjadi kode biner kemudian mengubah angka warna dari masing-masing piksel gambar menjadi kode biner kemudian di setiap bit ke delapan diubah atau disisipkan dengan 1 bit pesan.

Penelitian yang ketiga dilakukan oleh Andrew Chandra dkk. (2015), dengan judul implementasi steganografi dengan metode *End of File* pada teks yang terenkripsi menggunakanRC6 ke dalam citra. Mengenai penelitian ini proses enkripsi RC6 dan steganografi *End of File* melakukan penyisipan pesan dengan baik karena ukuran citra penampang dapat bertambah sesuai dengan pesan yang ingin disisipkan, dengan mnggunakan metode *End of File* maka pesan yang disisipkan tidak dibatasi perubahan ukuran *file* citra ini tergantung dari besarnya citra yang digunakan dan juga besarnya pesan yang digunakan dan panjang kunci yang disisipkan pasti sebesar 16 byte dikarenakan melalui proses enkripsi MD5.

**BAB II**

**LANDASAN TEORI**

* 1. **Pesan Teks**

Pesan adalah sesuatu yang disampaikan oleh komunikator kepada komunikan melalui proses komunikasi (Tasmara, 1987). Sedangkan pesan dalam buku pengantar Ilmu Komunikasi yang ditulis oleh Hafied (2004), bahwa pesan adalah serangkaian isyarat/symbol yang diciptakan oleh seseorang untuk maksud tertentu dengan harapan bahwa penyampaian isyarat/simbol itu akan berhasil dalam menimbulkan sesuatu.

Menurut Kamus Besar Bahasa Indonesia teks didefinisikan sebagai naskah yang berupa kata-kata asli dari pengarang. Sedangkan menurut Luxemburg dkk. (1989), teks adalah sebuah ungkapan yang memiliki isi, sintaksis pragmatik, dan itu semua menjadi sebuah kesatuan yang menjadikan teks memiliki makna yang menjadi sebuah penjelasan akan sebuah hal.

Berdasarkan uraian di atas pesan teks merupakan suatu naskah yang disampaikan melalui proses komunikasi.

* 1. **Kriptografi**
     1. **Definisi kriptografi**

Kriptografi berasal dari Bahasa Yunani, *Crypto* dan *Graphia*. *Crypto* berarti *secret* (rahasia) dan *Graphia* berarti *writing* (tulisan). Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengn sidik jari digital (Ariyus, 2005).

Kriptografi adalah suatu metode keamanan untuk melindungi suatu informasi dengan menggunakan kata-kata sandi yang hanya bisa dimengerti oleh orang yang berhak mengakses informasi untuk melindungi informasi yang melalui jaringan komunikasi yang menggunakan *landline* (kabel bawah tanah), satelit komunikasi, dan fasilitas *microwafe* (gelombang mikro). Prosedur-prosedur kriptografi juga bisa digunakan untuk authentikasi pesan, digital *signature*, dan dan indentifikasi pribadi untuk mengotorisasi transfer uang secara digital melalui ATM, kartu kredit yang melalui suatu jaringan. Kriptografi juga merupakan suatu metode yang digunakan untuk melindungi berbagai macam data yang prosesnya disebut dengan *encryption*, yaitu suatu proses yang mengkonversi sebuah pesan awal *plaintext* menjadi sebuah *chipertext* yang bisa dibalik ke bentuk asli seperti semula, yang juga bisa disebut sebagai proses *decoding* atau *decryption* (Ariyus, 2008).

* + 1. **Sejarah kriptografi**

Ilmu kriptografi sebenarnya sudah mulai dipelajari manusia sejak tahun 400-SM, yaitu pada zaman Yunani Kuno. Bidang ilmu ini terus berkembang seiring dengan kemajuan peradaban manusia, dan memegang peranan penting dalam strategi peperangan yang terjasi dalam sejarah manusia, mulai dari sistem kriptografi *“Caesar Chiper”* yang terkenal pada zaman Romawi Kuno, “*Playfair* Chiper” yang digunakan Inggris dan “ADFVGX Chiper” yang digunakan Jerman pada Perang Dunia I, hingga algoritma-algoritma kriptografi rotor yang popular pada Perang Dunia II, seperti Sigaba / M-134 (Amerika Serikat), Typex (Inggris), Purple (Jepang), dan mesin kriptografi lagendaris Enigma (Jerman) (Raharjo, 1998).

Induk dari keilmuan kriptografi sebenarnya adalah matematika, khususnya teori aljabar yang mendasar pada ilmu bilangan. Oleh karena itu, kriptografi semakin berkembang ketika computer ditemukan. Sebab dengan penemuan computer memungkinkkan dilakukannya perhitungan yang rumit dan kompleks dalam waktu yang relatif sangat singkat, dari suatu hal yang sebelumnya tidak dapat dilakukan. Dari hal tersebut lahirlah banyak teori dan algoritma penyandian data yang semakin kompleks dan sulit dipecahkan.

* + 1. **Tujuan kriptografi**

Tujuan kriptografi meliputi empat aspek penting, yaitu sebagai berikut (Kurniawan, 2004) :

1. Kerahasiaan

Kerahasiaan bertujuan untuk menjaga agar informasi atau pesan yang ada tidak dapat dibaca oleh pihak-pihak yang tidak berhak.

1. Integritas Data

Integritas data bertujuan untuk menjamin bahwa pesan masih asli atau belum pernah dimanipulasi oleh pihak yang tidak berhak selama proses pegiriman.

1. Otentikasi

Otentikasi bertujuan untuk mengidentifikasi tentang kebenaran dan keaslian pesan dari pihak-pihak yang berkomunikasi.

1. Anti Penyangkalan

Anti penyangkalan bertujuan untuk mencegah pihak yang mengirim pesan melakukan penyangkalan terhadap pesan yang dikirimnya.

* 1. **RC6 *Block Cipher***

RC6 merupakan algoritma yang merupakan keturunan dari RC5 yang juga merupakan kandidat AES *(Advanced Encryption Standard).* Pada mulanya, perancangan RC6 diawali ketika RC5 dianggap dapat dijadikan kandidat untuk mengikuti kompetisi pemilihan AES. Modifikasi kemudian dibuat untuk meningkatkan keamanan dan performa dan juga untuk dapat memenuhi persyaratan AES.

RC6 dirancang untuk menghilangkan segala ketidakamanan yang ditemukan pada RC5, karena analisis pada RC5 menunjukkan bahwa ternyata jumlah rotasi yang terjadi pada RC5 tidak sepenuhnya bergantung pada data yang terdapat dalam blok. Selain itu, serangan kriptanalisis diferensial juga ternyata dapat menembus keamanan yang ditawarkan RC5

RC6 juga dirancang untuk memenuhi persyaratan AES yang diantaranya adalah kemampuan untuk beroperasi pada mode blok 128 bit. Jika besar blok 128 bit langsung dipaksakan untuk diimplementasikan dengan algoritma RC5, maka akan dibutuhkan register kerja 64 bit. Spesifikasi arsitektur dan bahasa yang menjadi tempat implementasi algoritma yang ditentukan oleh AES belum mendukung pengoperasian 64 bit yang efisien. Oleh karena itu, daripada menggunakan 2 *register* 64 bit seperti pada RC5, RC6 menggunakan 4 register 32 bit. Karena menggunakan 4 register maka akan terdapat 2 operasi rotasi pada setiap *half-round* yang ada, dan juga akan lebih banyak bit-bit yang akan digunakan untuk mempengaruhi banyaknya bit yang dirotasi.

RC6 seperti juga RC5 mengeksploitasi penggunaan operasi-operasi primitif yang diimplementasikan secara efisien dalam prosesor-prosesor modern. RC6 juga selain menggunakan ketiga operasi primitif yang digunakan dalam RC5, juga menggunakan operasi perkalian 32-bit yang telah diimplementasikan secara efisien dalam prosesor modern saat ini. Primitif operasi perkalian ini sangat efektif dalam menghasilkan efek *“diffusion”* atau penyebaran yang tentu saja mengakibatkan RC6 lebih aman daripada RC5. Operasi perkalian ini digunakan untuk menghitung jumlah bit yang dirotasi sehingga konsep data-*dependent rotations* dapat dengan lebih sempurna diimplementasikan.

* + 1. **Operasi – operasi primitif**

Selain ketiga operasi primitif yang digunakan pada RC5, RC6 juga menggunakan operasi perkalian modulo-2w.

Berikut adalah daftar operator-operator primitif (termasuk invers) yang digunakan dalam RC6:

:

:

:

:

: dari

* + 1. **Algoritma RC6**

Algoritma RC6 seperti juga RC5 merupakan algoritma *cipher* yang terparameterisasi. RC6 secara tepat ditulis sebagai:

Nilai parameter w, r, dan b menyatakan hal yang sama seperti yang ditunjukkan dalam algoritma RC5. Algoritma RC6 yang dipakai sebagai kandidat AES adalah RC6-32/20/b, yang berarti ukuran *word* 32 bit, jumlah ronde 20 kali, dengan panjang kunci b ditentukan pengguna.

* + 1. ***Key expansion algorithm***

Algoritma untuk membangkitkan kunci internal sama seperti pada RC5. Nilai konstanta Pw dan Qw yang digunakan juga sama, tetapi ukuran array S tidak sama dengan yang seperti RC5. Ukuran t dari array S dalam RC6 adalah , yang berarti terdapat lebih banyak kunci internal yang dibangkitkan daripada jumlah kunci internal RC5. Berikut algoritmanya:

S[0] = Pw

**for** i = 1 to (2r + 3) **do**

S[i] = S[i – 1] + Qw

**End**

i = 0

j = 0

A = B = 0

**for** 3 × max(c, (2r + 4)) **times do**

S[i] = (S[i] + A + B) <<< 3

A = S[i] L[i] = (L[j] + A + B) <<< 3

B = L[i]

i = (i + 1) mod (2r+4)

j = (j + 1) mod c

* + 1. **Algoritma enkripsi**

Fungsi enkripsi menerima *input* 1 blok *plaintext* yang terbagi dalam 4 register yang masing-masing berupa w-bit *word*, yaitu A, B, C, dan D. *Ciphertext* hasil proses terbagi dan disimpan dalam A, B, C, dan D. Dalam proses enkripsi diperlukan tabel kunci S yang dianggap telah didapat dari proses sebelumnya.

Secara lebih detil, proses enkripsi dengan RC6 dapat dibagi dalam beberapa langkah. Dalam penjelasan berikut, notasi (A,B,C,D) = (B,C,D,A) berarti adalah operasi *assignment* yang dilakukan paralel (bersamaan) untuk setiap elemen di ruas kanan ke ruas kiri yang berkorespondensi. Langkah-langkahnya adalah sebagai berikut:

1. Mula-mula lakukan *half-round loop* yang seperti pada RC5:

**for** i = 1 to r **do**

A = ((AB) <<< B) + S[i] ⊕

(A,B) = (B,A)

1. Lakukan dua proses RC5 secara paralel, yang satu untuk register A, B dan yang lain untuk register C, D

**for** i = 1 to r **do**

A = ((AB) <<< B) + S[2i] ⊕

C = ((CD) <<< D) + S[2i+1] ⊕

(A,B) = (B,A)

(C,D) = (D,C)

1. Pada tahap pertukaran, daripada menukar A dengan B, dan C dengan D, lakukan permutasi antar keempat register (A,B,C,D) = (B,C,D,A), sehingga komputasi AB bercampur dengan komputasi CD.

**for** i = 1 to r **do**

A = ((AB) <<< B) + S[2i] ⊕

C = ((CD) <<< D) + S[2i+1] ⊕

(A,B,C,D) = (B,C,D,A)

1. Campurkan komputasi AB dengan CD lebih jauh, yaitu dengan mempertukarkan kedua nilai yang menyatakan jumlah rotasi pada masing-masing komputasi.

**fori** = 1 to r **do**

A = ((A⊕B) <<< D) + S[2i]

C = ((C⊕D) <<< B) + S[2i+1]

(A,B,C,D) = (B,C,D,A

1. Daripada menggunakan nilai B dan D secara langsung, RC6 menggunakan hasil transformasi kedua register ini. Hal ini dilakukan untuk tidak mengulangi masalah rotasi seperti pada RC5 di mana tidak seluruh bit dalam data yang berpengaruh dalam rotasi. Oleh karena itu, fungsi transformasi yang dipilih harus dapat memanfaat seluruh bit di dalam data untuk mengatur jumlah bit yang dirotasikan. Fungsi yang dipilih adalah yang kemudian diikuti dengan rotasi ke kiri sebanyak 5 bit. Transformasi ini terpilih karena fungsi f(x) yang merupakan fungsi satu-ke-satu memiliki bit-bit orde atas yang menentukan jumlah rotasi yang akan digunakan yang sangat bergantung pada x.

**for** i = 1 to r **do**

p = (B × (2B + 1)) <<< 5

q = (D × (2D + 1)) <<< 5

A = ((A⊕p) <<< q) + S[2i]

C = ((C⊕q) <<< p) + S[2i+1]

(A,B,C,D) = (B,C,D,A)

1. Setelah *loop* di atas selesai, akan terdapat hasil di mana *plaintext* bisa menunjukkan bagian input ronde pertama dalam enkripsi dan *ciphertext* bisa menunjukkan bagian input ronde terakhir dalam enkripsi. Oleh karena itu perlu ditambahkan langkah-langkah di awal dan di akhir *loop* untuk menyamarkan hubungan ini. Sehingga, terbentuklah algoritma enkripsi RC6 yang sebagai berikut:

B = B + S[0]

D = D + S[1]

**for** i = 1 to r **do**

p = (B × (2B + 1)) <<< 5

q = (D × (2D + 1)) <<< 5

A = ((A⊕p) <<< q) + S[2i]

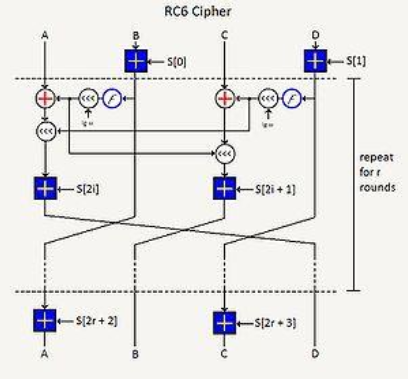
C = ((C⊕q) <<< p) + S[2i+1]

(A,B,C,D) = (B,C,D,A)

A = A + S[2r + 2]

C = C + S[2r + 3]

Perlu diketahui juga, dalam varian baru RC6 jumlah rotasi ke kiri yang mengikuti fungsi kuadrat bukan 5 bit tetapi adalah



**Gambar 2.3. Diagram Enkripsi RC6**

* + 1. **Algoritma dekripsi**

Sama seperti pada RC5, algoritma dekripsi RC6 juga merupakan penurunan dari algoritma enkripsi. Algoritmanya sebagai berikut:

C = C – S[2r + 3]

A = A – S[2r + 2]

**for** i = r downto 1 **do**

(A,B,C,D) = (D,A,B,C)

p = (D × (2D + 1)) <<< 5

q = (B × (2B + 1)) <<< 5

C = ((C – S[2i + 1]) >>> q)p ⊕

A = ((A – S[2i]) >>> p)q ⊕

D = D – S[1]

B = B – S[0]

* 1. **Steganografi**

Steganografi *(steganography)* adalah ilmu, teknik atau seni menyembunyikan pesan rahasia (hiding message) atau tulisan rahasia *(covered writing)* sehingga keberadaan pesan tidak terdeteksi orang lain kecuali pengirim dan penerima pesan tersebut. Steganografi berasal dari bahasa Yunani yaitu steganos (tersembunyi/menyembunyikan) dan graphy (tulisan), sehingga secara lengkap bermakna tulisan yang disembunyikan.

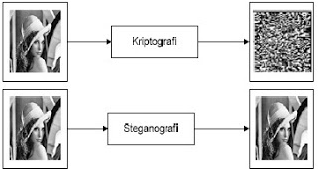
Berikut ini beberapa pengertian Steganografi dari beberapa sumber buku:

1. Menurut Arubusman (2007), Steganografi adalah suatu teknik untuk menyembunyikan informasi pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya.
2. Menurut Munir (2009), Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia *(hiding message)* sedemikian sehingga keberadaan *(eksistensi)* pesan yang tidak terdeteksi oleh indera manusia.
3. Menurut Ariyus (2006), steganografi merupakan cabang ilmu yang mempelajari bagaimana menyimpan informasi rahasia di dalam informasi lainnya (Muchlisin Riadi, 2017)
   * 1. **Perbedaan steganografi dan kriptografi**

Terdapat perbedaan antara steganografi dengan kriptografi. Perbedaan terletak pada visibilitas pesan, pada kriptografi pihak ketiga dapat mendeteksi adanya data acak *(chipertext),* karena hasil dari kriptografi berupa data yang berbeda dari bentuk aslinya dan biasanya datanya seolah-olah berantakan, tetapi dapat dikembalikan ke bentuk semula.

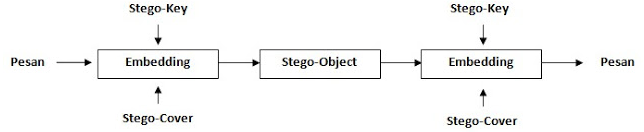
Berbeda dengan kriptografi yang menjaga kerahasian pesan dengan cara mengubah bentuk pesan agar tidak dapat dipahami oleh orang lain, steganografi merupakan suatu teknik penyembunyian pesan pada suatu medium. Perlu diperhatikan dalam steganografi, suatu pesan tidak harus diubah, tetapi pesan tersebut disembunyikan pada suatu medium agar pesan tersebut tidak terlihat.

Salah satu keuntungan steganografi dibandingkan dengan kriptografi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung pesan tidak menimbulkan kecurigaan bagi pihak ketiga. Gambar berikut ini menggambarkan ilustrasi perbedaan steganografi dengan kriptografi.



**Gambar 2.4. Perbedaan Steganografi dan Kriptografi**

* + 1. **Cara kerja steganografi**

Untuk menyisipkan data yang ingin disembunyikan membutuhkan dua unsur. Unsur pertama adalah media penampung seperti citra, suara, video dan sebagainya yang terlihat tidak mencurigakan untuk menyimpan pesan rahasia. Unsur kedua adalah pesan yang ingin disembunyikan yaitu media penampungnya berupa citra yang disebut *cover-object* dan citra yang telah disisipi pesan disebut *stego-object.*

**Gambar 2.5. Cara Kerja Steganografi**

Secara umum, terdapat dua proses didalam steganografi, yaitu proses *embedding* untuk menyisipkan pesan ke dalam *cover-object* dan proses *decoding* untuk ekstraksi pesan dari *stego-object*. Kedua proses ini mungkin memerlukan kunci rahasia yang dinamakan *stego-key* agar hanya pihak yang berhak saja yang dapat melakukan penyisipan dan ekstraksi pesan.

* + 1. **Kriteria dan aspek steganografi**

Penyembunyian data rahasia ke dalam media digital mengubah kualitas media tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data diantaranya adalah:

1. *Fidelity*. Mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2. *Robustness*. Data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung (seperti pengubahan kontras, penajaman, pemampatan, penambahan *noise*, perbesaran gambar, pemotongan *(cropping),* enkripsi, dan sebagainya). Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak.
3. *Recovery*. Data yang disembunyikan harus dapat diungkapkan kembali *(recovery)*. Karena tujuan steganografi adalah data *hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

Sebuah steganografi memiliki tiga aspek yang dapat menentukan berhasil tidaknya atau baik-tidaknya sebuah steganografi dalam melakukan pekerjaannya (Ermadi dkk., 2004), yaitu:

1. Kapasitas *(capacity).* Kapasitas merujuk pada jumlah informasi yang bisa disembunyikan dalam *medium cover*. Keamanan adalah ketidakmampuan pengamat untuk mendeteksi pesan tersembunyi dan ketahanan adalah jumlah modifikasi *medium stego* yang bisa bertahan sebelum musuh merusak pesan rahasia yang tersembunyi tersebut.
2. Keamanan (*security).* Keamanan dari sistem steganografi klasik mewujudkan kerahasiaan sistem *encoding*-nya. Teori informasi memungkinkan kita untuk lebih spesifik pada apa yang dimaksudkan dengan suatu sistem yang benar-benar aman.
3. Ketahanan *(robustness).* Ketahanan mengacu pada data citra penampang (seperti pengubahan kontras, penajaman, rotasi, perbesaran gambar, pemotongan dan sebagainya). Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak.
   * 1. **Jenis – jenis teknik steganografi**

Berdasarkan teknik steganografi yang digunakan terdapat tujuh jenis teknik steganografi, yaitu sebagai berikut (Ariyus, 2009):

1. *Injection. Injection* merupakan suatu teknik menanamkan pesan rahasia secara langsung ke suatu media. Salah satu masalah dari teknik ini adalah ukuran media yang diinjeksi menjadi lebih besar dari ukuran normalnya sehingga mudah dideteksi. Teknik ini sering juga disebut embedding.
2. Substitusi. Data normal digantikan dengan data rahasia. Biasanya, hasil teknik ini tidak terlalu mengubah ukuran data asli, tetapi tergantung pada *file* media dan data yang akan disembunyikan. Teknik substitusi bisa menurunkan kualitas media yang ditumpangi.
3. Transformasi Domain. Teknik ini sangat efektif. Pada dasarnya, transformasi domain menyembunyikan data pada *transform space.*
4. *Spread Spectrum. Spread spectrum* merupakan teknik pentransmisian menggunakan *pseudo-noise code*, yang independen terhadap data informasi sebagai modulator bentuk gelombang untuk menyebarkan energi sinyal dalam sebuah jalur komunikasi *(bandwith)* yang lebih besar daripada sinyal jalur komunikasi informasi. Oleh penerima, sinyal dikumpulkan kembali menggunakan replika *pseudo-noise code* tersinkronisasi.
5. *Statistical Method*. Teknik ini disebut juga skema *steganographic* 1 bit. Skema tersebut menanamkan satu bit informasi pada media tumpangan dan mengubah statistik walaupun hanya 1 bit. Perubahan statistik ditunjukkan dengan indikasi 1 dan jika tidak ada perubahan, terlihat indikasi 0. Sistem ini bekerja berdasarkan kemampuan penerima dalam membedakan antara informasi yang dimodifikasi dan yang belum.
6. *Distortion.* Metode ini menciptakan perubahan atas benda yang ditumpangi oleh data rahasia.
7. *Cover Generation.* Metode ini lebih unik daripada metode lainnya karena *cover object* dipilih untuk menyembunyikan pesan.
   1. **LSB *(Least Significant Bit)***

Metode LSB *(Least Significant Bit)* merupakan metode steganografi yang paling sederhana dan paling mudah dimplemetasikan. Untuk menjelaskan metode ini digunakan citra digital sebagai *covertext*. Setiap pixel di dalam citra berukuran 1 sampai 3 byte. Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), ada bit yang paling berarti *(most significant bit atau MSB)* dan bit yang paling kurang berarti *(leastsignificant bit atau LSB).*

Misalnya pada byte 11010010, bit 1 yang pertama (digarisbawahi) adalah bit MSB dan bit 0 yang terakhir (digarisbawahi) adalah bit LSB. Bit yang cocok untuk diganti dengan bit pesan adalah bit LSB, sebab modifikasinya hanya mengubah nilai byte tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalnya byte tersebut di dalam citra memberikan persepsi warna merah, maka perubahan satu bit LSB hanya mengubah persepsi warna merah tidak terlalu berarti. Mata manusia tidak dapat membedakan perubahan sekecil ini (Munir, 2006).

Sebagai ilustrasi, misalkan segmen pixel-pixel citra sebelum disisipkan pesan adalah

00110011 10100010 11100010 01101111

dan misalkan pesan rahasia (yang telah dikonversi ke biner) adalah 0110. Setiap bit pesan menggantikan posisi LSB dari segmen pixel-pixel citra menjadi:

00110010 10100011 11100011 10010000

Untuk membuat pesan tidak dapat dilacak, bit-bit pesan tidak *mengganti byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit LSB-nya dipilih secara acak, misalnya byte nomor 36, 5, 21, 10, 18, 49 (Munir, 2006).

* + 1. **Ukuran teks yang disembunyikan**

Ukuran maksimal teks yang dapat disembunyikan bergantung pada ukuran citra penampung yang digunakan. Semakin besar citra yang digunakan maka akan semakin banyak pula jumlah karakter yang dapat disembunyikan. Rumus untuk menghitung jumlah maksimal karakter yang dapat disembunyikan pada citra adalah

Misalnya citra yang digunakan sebagai wadah penampung berukuran 200 × 200, artinya citra ini mempunyai panjang 200 pixel dan lebar 200 pixel, sehingga total pixel citra tersebut adalah 40.000 pixel, karena 1 karakter terdiri dari 8 bit maka karakter disembunyikan pada setiap 8 pixel sehingga maksimal karakter yang dapat disemunyikan adalah sebanyak .

Sedangkan untuk citra 24 bit yang berukuran 256 × 256, terdapat 65.536 pixel. Karena setiap pixel berukuran 3 *byte* (komponen RGB), berarti seluruhnya ada 65.536 × 3 = 196.608 byte. Setiap *byte* hanya bisa menyembunyikan satu bit maka maksimal karakter yang dapat disembunyikan adalah karakter (Munir, 2006).

* 1. **Bahasa Pemrograman *Hypertext Preprocessor***

PHP Kepanjangan dari PHP adalah "*Hypertext Preprocessor*" (ini merupakan singkatan rekursif).PHP adalah bahasa scriptingweb *HTML-embedded*. Ini berarti kode PHP dapat disisipkan ke dalam HTML halaman Web. Ketika sebuah halaman PHP diakses, kode PHP dibaca atau"diurai" oleh *server*. *Output* dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dibuka, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses *database* dan informasi aman lainnya. (Ferdianto, 2013)

Banyak sintaks PHP yang hasil adaptasi dari bahasa lain seperti bahasa C, Java dan Perl. Namun, PHP memiliki sejumlah fitur unik dan fungsi tertentu juga. Tujuan dari bahasa pemrograman PHP adalah untuk memungkinkan pengemban web untuk menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah. PHP juga bagus untuk menciptakan situs Web *database-driven*. Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya yaitu PHP.net. (Ferdianto, 2013)

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukansebuah kompilasi dalam penggunaanya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulaiapache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudahkarena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime*melalui *console* serta juga dapat menjalankan perintah-perintah sistem. (Ferdianto, 2013)
   1. ***Hypertext Markup Language* (HTML)**

HTML adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman *web*. HTML dapat dibaca oleh berbagai *platform* seperti : Windows, Linux, Macintosh. Kata *Markup Language* pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana pengguna bisa mengatur judul, garis, tabel, gambar, dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML. HTML sendiri dikeluarkan oleh W3C (*Word Wide Web Consortium*), setiap terjadi perkembangan level HTML harus dievakuasi ketat dan disetujui oleh W3C. Contoh tag HTML antara lain : <head>, <body> dan <table> (Kadir, 2002).

* 1. ***Cascading Style Sheet* (CSS)**

*Cascading Style Sheets* (CSS) adalah salah satu bahasa pemrograman desain *web* (*style sheetlanguage*) yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan bahasa penanda (*markup language*). Biasanya CSS digunakan untuk mendesain sebuah malam HTML dan XHTML, tetapi sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen *XML*,termasuk SVG dan XUL. CSS dibuat untuk memisahkan kontek utama(biasanya dibuat dengan menggunakan bahasa HTMLdan sejenisnya) dengan tampilan dokumen yangmeliputi *layout*, warna dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada *web*, menyediakan lebih banyak fleksibilitas dan control dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi banyak halaman untuk sebuah formating dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tables pada layout desain *web* (*layout* tanpa tabel)

*Style Sheet* adalah sebuah *text file*yang sederhana (dimana berekstensi \*.css), ditulis menurut aturan bahasa yang dipaparkan pada rekomendasi CSS1 atau CSS2. Cara kerja CSS dengan menggunakan dua buah elemen penting untuk pemformatan tampilannya, diantaranya selektor dan deklarator. Dua buah elemen ini berfungsi sebagai penentu format tampilan dan lainnya menempatkan format tampilan tersebut. Deklarator berisi beberapa perintah-perintah CSS untuk menentukan format dari sebuah elemen pada halaman *web*. Sedangkan selektor adalah sebuah perintah lanjut dari deklarator dan berfungsi menempatkan format tampilan dari deklarator.

Dalam *Cascading Style Sheets* ada dua cara menghubungkan sebuah dokumen HTML dengan CSS

*1. Selector class*

*2. Selector ID*

*3. Selector Descendant* (turunan)

*4. Selector Link Pseudo Class*

*5. Selector Pseudo elemen*

*6. Selector dynamic pseudo class*

*7. Selector languange*

*8. Selector child*

*9. Selector first-child*

10. *Selector adjacent* (berdekatan)

* 1. ***Rational Unified Process* (RUP)**

*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah

menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perankat lunak. Gambar dibawah menunjukkan secara keseluruhan arsitektur yang dimiliki RUP.

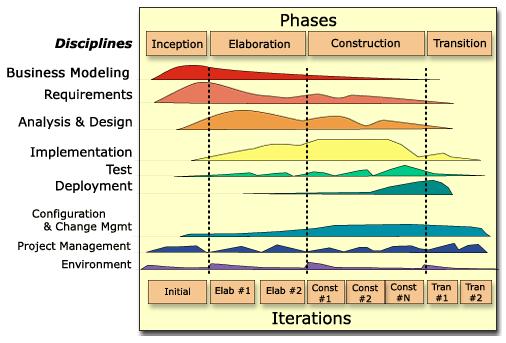
Ada beberapa keuntungan dengan mengunakan RUP diantaranya :

1. Menyediakan akses yang mudah terhadap pengetahuan dasar bagi anggota tim.
2. Mendukung proses pengulangan dalam pengembangan software.
3. Memungkinkan adanya penambahan-penambahan pada proses.
4. Memungkinkan untuk secara sistematis mengontrol perubahan - perubahan yang terjadi pada software selama proses pengembangannya.

Kekurangan Pengembangan Perangkat Lunak RUP adalah metodologi ini hanya dapat digunakan pada pengembangan perangkat lunak yangberorientasi objek dengan berfokus pada UML *(Unified Modeling Language).*

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui Gambar 2.5 dapat dilihat bahwa RUP memiliki 2 dimensi yaitu:

1. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari fase selanjutnya. Setiap fase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction*, dan *Transition.*
2. Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Manegement, Project Management, Environtment.*



**Gambar 2.5 Arsitektur *Rational Unified Process***

Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak yaitu:

1. *Inception*

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*). Pada akhir fase ini, prototipe perangkat lunak versi *Alpha* harus sudah dirilis.

1. *Elaboration*

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilisan prototipe versi *Betha* dari perangkat lunak.

1. *Construction*

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta  dokumentasi perangkat lunak.

1. *Transition*

Instalasi *deployment* dan sosialisasi perangkat lunak dilakukan pada tahap ini. Pada tahapan ini dilakukan tahap evaluasi dari sistem yang telah direncanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari sistem yang telah dibangun.

* 1. ***Flowchart***

*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya. Simbol-simbol *flowchart* adalah standar yang ditentukan oleh *Amerika National Standard Institute Inc*. Simbol–simbol yang digunakan dalam *flowchart* ditampilkan pada Tabel 2.1.

**Tabel. 2.1 Simbol-simbol *flowchart***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO.** | **SIMBOL** | **NAMA** | **KETERANGAN** |
| 1. |  | Proses | Mempresentasikan operasi. |
| 2. |  | *Input / Output* | Mempresentasikan *Input* data atau *Output* data yang diproses atau informasi. |
| 3. |  | Keputusan | Keputusan dalam program. |
| 4. |  | Dokumen | Dokument I / O dalam format cetak. |
| 5. |  | *Terminal points* | Awal / akhir *flowchart*. |
| 6. |  | *Preparation* | Pemberian harga awal. |

* 1. ***Unified Modeling Language* (UML)**

*Unified Modeling Language* (UML) merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA&D) yang dimunculkan sepenggunar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. UML merupakan bahasa standar untuk penulisan *blue print software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak (Rosa dkk., 2011).

Dalam UML terdapat diagram-diagram yang bisa menggambarkan bagian atau aspek tertentu dari sebuah sistem. Sehingga dapat terlihat jelas alur dan gambaran umum dari perangkat lunak yang dibangun. Ada beberapa jenis diagram dalam UML (Rosa dkk., 2011), yaitu:

1. ***Use case* *Diagram***

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapatjuga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem (Rosa dkk., 2011).

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabel 2.2 Simbol *Use Case Diagram*** | | | |
| **NO** | **SIMBOL** | **NAMA** | **KETERANGAN** |
| 1. |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case* |
| 2. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (dependent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent) |

**Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SIMBOL** | **NAMA** | **KETERANGAN** |
| 3. |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*) |
| 4. |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara eksplisit |
| 5. |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan |
| 6. |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7. |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas |
| 8 |  | *Use case* | Deskripsi dari uraian aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 9. |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi) |
| 10. |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

1. ***Class Diagram***

Menggambarkan struktur statis *class* di dalam sistem. *Clas*s merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram* (Rosa dkk., 2011).

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabel 2.3 Simbol *Class Diagram*** | | | |
| **NO** | **SIMBOL** | **NAMA** | **KETERANGAN** |
| 1. |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |
| 2. |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek |
| 3. |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4. |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 5. |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek |
| 6. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |

1. ***Sequence Diagram***

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksiantara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Rosa dkk., 2011).

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabel 2.4 Simbol *Sequence Diagram*** | | | |
| **NO** | **SIMBOL** | **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 |
| 4 |  | *Message* | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy* |

1. ***Activity Diagram***

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi (Rosa dkk., 2011).

**Tabel 2.5 Simbol *Activity Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |

**Tabel 2.5 Simbol *Activity Diagram* (Lanjutan)**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |
| **6** |  | *Line Conector* | Digunakan untuk menghubungkan satu simbol dengan simbol yang lain |
| **7** |  | *Decision* | Menunjukan suatu keputusan yang mempunyai satu atau lebih transisi sesuai dengan kondisi |

**BAB III**

**METODOLOGI PENELITIAN**

* 1. **Waktu dan Tempat Penelitian**

**3.1.1 Waktu**

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Januari 2019 sampai dengan Maret 2019. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

**Tabel 3.1** ***Gannt Chart* Waktu Penelitian**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Uraian** | **Waktu (2019)** | | | | | | | | | | | | | | | |
| **Mei** | | | | **Juni** | | | | **Juli** | | | | **Agustus** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** | *Inception* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | *Elaboration* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | C*onstruction* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | *Transition* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**3.1.2 Tempat penelitian**

Adapun tempat penelitian tugas akhir yang akan dilakukan tidak mengacu pada tempat tertentu

* 1. **Metode Pengumpulan Data**

Dalam penelitian ini metode pengumpulan data yang digunakan pada perancangan aplikasi tersebut adalah studi *literatur*. Metode ini dilaksanakan dengan melakukan studi kepustakaan yang relevan. Metode ini dilakukan untuk mencari sumber pelengkap yang berhubungan dengan aplikasi yang akan dibangun, yaitu dengan mencari referensi yang berkaitan dengan kata serapan, sehingga dapat di implementasikan dalam aplikasi tersebut, mulai dari buku-buku, jurnal maupun artikel dan sumber-sumber lain di *internet*.

* 1. **Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process* (RUP). Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak, yaitu:

* + 1. ***Inception***

Pada *fase* ini dilakukan proses pengidentifikasian sistem, dilakukan dengan analisis kebutuhan akan aplikasi, melakukan kajian terhadap penelitian yang terkait dengan algoritma *Rivest Code* *6* (RC6) dan algoritma *Least Significant* *Bit* (LSB)

* + 1. ***Elaboration***

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan *flowchart* meliputi *flowchart* algoritma RC6 dan algoritma LSB. Pada perancangan ini, digunakan juga UML (*Unified Modelling Language*) yang meliputi *activity diagram, class diagram, sequence diagram* dan *use case diagram*.

* + 1. ***Construction***

Proses yang dilakukan pada tahap ini yaitu membangun aplikasi dengan perancangan yang telah dilakukan sebelumnya, mulai dari tampilan *interface* menu *home*, proses enkripsi dan dekripsi serta tampilan proses penyisipan dan ekstraksi. Proses yang juga dilakukan pada tahap ini yaitu penerapan *coding* algoritma RC6 dan algoritma LSB pada aplikasi.

* + 1. ***Transition***

Menguji dan melakukan dokumentasi pada aplikasi yang telah dibuat dan memperbaiki masalah-masalah yang muncul saat dan setelah pengujian.