



**PENGAMANAN CITRA DIGITAL DENGAN METODE F5  
MENGUNAKAN VISUAL STUDIO 2010**

**Disusun untuk Memenuhi Persyaratan Ujian Akhir Memperoleh Gelar  
Sarjana Komputer pada Fakultas Sains dan Teknologi  
Universitas Pembangunan Panca Budi  
Medan**

**SKRIPSI**

**OLEH:**

**NAMA : FIRNANDO HANSGITO SILALAHIS**

**N.P.M : 1624371090**

**PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS PEMBANGUNAN PANCA BUDI**

**MEDAN**

**2022**

PENGESAHAN TUGAS AKHIR

JUDUL : PENCAMANAN CITRA DIGITAL DENGAN METODE F5 MENGGUNAKAN VISUAL STUDIO 2010

NAMA : FIRNANDO HANSGITO SILALAH S  
N.P.M : 1624371090  
FAKULTAS : SAINS & TEKNOLOGI  
PROGRAM STUDI : Sistem Komputer  
TANGGAL KELULUSAN : 01 Desember 2022



DEKAN



Hamdani, ST., MT.

KETUA PROGRAM STUDI



Eko Hariyanto, S.Kom., M.Kom

DISETUJUI  
KOMISI PEMBIMBING

PEMBIMBING I



Eko Hariyanto, S.Kom., M.Kom

PEMBIMBING II



Nova Mayasari, S.Kom., M.Kom.

## SURAT PENYATAAN

Yang bertandatangan dibawah ini:

Nama : Firnando Hansgito Silalahi S

NPM : 1624371090

Prodi : Teknik Elektro

Fakultas : Sains dan Teknologi

Tahun Lulus : 2022

Judul Skripsi : Perancangan Citra Digital dengan Metode F5 Menggunakan Visual Studio 210

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya buka hasil plagiat
2. Saya tidak akan menuntut perbaikan nilai Indeks Prestasi (IPK) setelah ujian sidang meja hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga dan saya tidak akan menuntut akibat publikasi tersebut.

Demikian pernyataan ini saya buat dengan sebenar-benarnya, diucapkan terimakasih.

YAYASAN PROF. DR. H. KADIRUN YAHYA

Medan, 01 Desember 2022

Yang membuat pernyataan,



Firnando Hansgito Silalahi S

## SURAT ORISINALITAS

Dengan ini menyatakan bahwa dalam skripsi ini tidak terdapat karya yang diajukan untuk memperoleh gelar sarjana didalam perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang tertulis di acu dalam skripsi ini dan disebutkan dalam daftar pustaka.

Medan, 01 Desember 2022  
Yang membuat pernyataan,



Firmando Hansgito Silalahi S

YAYASAN PROF. DR. H. KADIRUN YAHY

## ABSTRAK

FIRNANDO HANSGITO SILALAH S

Pengamanan Citra Digital Dengan Metode F5 Menggunakan Visual Studio 2010  
2022

keamanan dalam pertukaran data sangatlah penting. komunikasi digital rentan terhadap pencurian informasi baik itu secara langsung ataupun tidak langsung. Hal ini dapat diatasi dengan beberapa cara yaitu dengan teknik untuk pengamanan data diantaranya kriptografi dan steganografi. Metode F5 merupakan metode yang menyisipkan bit data pesan ke dalam bit hasil kuantisasi yang terlebih dahulu telah dipermutasi. Metode F5 mempunyai keakuratan penyembunyian pesan teks dan keakuratan descriptory cukup baik. Pada skripsi telah dilakukan implementasi metode F5 untuk menyisipkan pesan ke dalam citra BMP, hasil implementasi kemudian dilakukan pengujian kapasitas pesan yang disisipkan terhadap banyaknya variasi warna pada file citra BMP. Selain itu dilakukan penyisipan pesan dan citra lain (BMP) kedalam file citra carier. Hasil aplikasi sistem keamanan data untuk memberikan kemudahan mengenai pengamanan *stegno* agar sistem keamanan data ini dapat berjalan dengan sempurna, pertama sekali harus ada *file* gambar yang ingin di enkripsikan selanjutnya jalankan aplikasi yang penulis rancang.

**Kata Kunci :** Citra, Metode F5, BMP, Steganografi

## KATA PENGANTAR

Puji dan syukur saya naikkan kehadirat Tuhan Yang Maha Esa yang senantiasa memberikan berkat yang berlimpah, sehingga skripsi ini dapat saya selesaikan dengan baik. Skripsi ini disusun untuk memenuhi salah satu persyaratan untuk memperoleh gelar Sarjana Komputer pada Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi.

Saya menyadari sepenuhnya bahwa skripsi ini masih memiliki banyak kekurangan, oleh karena itu saya sangat mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan skripsi ini.

Dalam penyelesaian skripsi ini, saya telah banyak menerima bantuan dan dorongan serta bimbingan dari berbagai pihak. Pada kesempatan ini, saya menyampaikan rasa terima kasih kepada:

1. Teristimewa kepada kedua orang tua yang telah mendukung penulis dalam doa dan memberikan semangat setiap harinya dalam penyusunan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M, sebagai Rektor Universitas Pembangunan Panca Budi Medan
3. Bapak Hamdani, S.T., M.T, sebagai Dekan Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan
4. Bapak Eko Hariyanto, S.Kom., M.Kom, selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan sekaligus Dosen Pembimbing I yang telah memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini
5. Ibu Nova Mayasari, S.Kom., M.Kom, selaku Dosen Pembimbing II yang telah memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini

6. Bapak dan Ibu Dosen dan Staf Fakultas Sains & Teknologi Universitas Pembangunan Panca Budi Medan
7. Seluruh rekan mahasiswa Fakultas Sains dan Teknologi dan teman-teman atas perhatian dan dorongan yang diberikan selama ini.

Akhir kata saya mengharapkan semoga skripsi ini dapat bermanfaat kepada para pembaca.

Medan, 01 Desember 2022

Penulis

**FIRNANDO HANSGITO SILALAHIS**  
**NPM. 1624371090**

## DAFTAR ISI

|                                 | Hal |
|---------------------------------|-----|
| ABSTRAK .....                   | i   |
| KATA PENGANTAR .....            | ii  |
| DAFTAR ISI.....                 | iv  |
| DAFTAR GAMBAR .....             | vii |
| DAFTAR TABEL.....               | ix  |
| BAB I PENDAHULUAN.....          | 1   |
| 1.1 Latar Belakang Masalah..... | 1   |
| 1.2 Rumusan Masalah .....       | 2   |
| 1.3 Batasan Masalah.....        | 2   |
| 1.4 Tujuan Penelitian .....     | 2   |
| 1.5 Manfaat Penelitian .....    | 3   |
| BAB II TINJAUAN PUSTAKA.....    | 4   |
| 2.1 Steganografi .....          | 4   |
| 2.2 Enkripsi .....              | 7   |
| 2.3 Dekripsi .....              | 9   |
| 2.4 Keamanan Data .....         | 11  |
| 2.5 Metode F5 .....             | 13  |
| 2.6 Perancangan .....           | 15  |

|   |           |
|---|-----------|
| 2.6.1 Cara Pendekatan.....                                      | 16        |
| 2.6.2 Prinsip Dan Petunjuk Perancangan .....                    | 18        |
| 2.6.3 Urutan Perancangan .....                                  | 20        |
| 2.7 Unified Modeling Language (UML).....                        | 21        |
| 2.7.1 Tujuan Pemanfaatan UML.....                               | 22        |
| 2.7.2 Struktur Diagram.....                                     | 23        |
| 2.8 Bahasa Pemrograman Visual Basic 2010 .....                  | 30        |
| <b>BAB III METODE PENELITIAN.....</b>                           | <b>31</b> |
| 3.1 Analisis Masalah.....                                       | 31        |
| 3.2 Strategi Pemecahan Masalah.....                             | 32        |
| 3.3 Analisa Kebutuhan <i>Hardware</i> Dan <i>Software</i> ..... | 32        |
| 3.4 Penerapan Metode F5.....                                    | 33        |
| 3.5 Desain Sistem.....  | 34        |
| 3.5.1 Use Case Diagram.....                                     | 34        |
| 3.5.2 Sequence Diagram .....                                    | 35        |
| 3.5.3 Activity Diagram.....                                     | 39        |
| 3.6 Rancangan Program .....                                     | 43        |
| 3.6.1 Rancangan <i>Output</i> .....                             | 43        |
| 3.6.2 Rancangan <i>Input</i> .....                              | 47        |
| <b>BAB IV HASIL DAN PEMBAHASAN .....</b>                        | <b>48</b> |

|  |        |
|--|--------|
| 4.1 Tampilan Hasil .....                                 | 48     |
| 4.1.1 Tampilan Menu Utama .....                          | 48     |
| 4.1.2 Tampilan <i>Encode</i> .....                       | 49     |
| 4.1.3 Tampilan <i>Form Decode</i> .....                  | 49     |
| 4.1.4 Tampilan Form Help .....                           | 50     |
| 4.1.5 Tampilan Form About.....                           | 51     |
| 4.2 Pengujian.....                                       | 52     |
| 4.2.1 Rencana Pengujian .....                            | 52     |
| 4.2.2 Pengujian Kasus dan Hasil.....                     | 53     |
| 4.3 Pembahasan.....                                      | 54     |
| 4.4 Kelebihan Dan Kekurangan Sistem Yang Dirancang ..... | 54     |
| <br>BAB V KESIMPULAN DAN SARAN.....                      | <br>55 |
| 5.1 Kesimpulan .....                                     | 55     |
| 5.2 Saran.....   | 55     |

DAFTAR PUSTAKA

## DAFTAR TABEL

|   | Hal |
|---|-----|
| Tabel 2.1 Simbol-Simbol <i>Use Case Diagram</i> ..... | 24  |
| Tabel 2.2 Simbol-Simbol <i>Sequence Diagram</i> ..... | 25  |
| Tabel 2.3 Simbol Activity Diagram .....               | 27  |
| Tabel 4.1 Skenario Pengujian Sistem.....              | 52  |
| Tabel 4.2 Pengujian Sistem data Keamanan Data .....   | 52  |



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi sekarang ini memungkinkan orang untuk dapat melakukan komunikasi maupun pertukaran data secara mudah. Karena itu tentunya keamanan data sangatlah penting, terutama dalam pekerjaan. Sebagai contoh, komunikasi *digital* rentan terhadap pencurian informasi baik itu secara langsung ataupun tidak langsung. Hal ini dapat diatasi dengan beberapa cara yaitu dengan teknik untuk pengamanan data diantaranya kriptografi dan steganografi. Kriptografi merupakan teknik pengenkripsian data berupa gambar sedangkan steganografi yaitu untuk mengamankan data dengan menyembunyikan isi pesan yang telah di kriptografi dalam suatu media gambar seperti *file* gambar yang berformat JPEG, sehingga hanya pihak terkait saja yang mengetahui adanya pesan rahasia dengan syarat pengirim dan penerima memasukkan *encode* yang sama.

Metode F5 merupakan metode yang menyisipkan bit data pesan ke dalam bit hasil kuantisasi yang terlebih dahulu telah dipermutasi. Metode F5 mempunyai keakuratan penyembunyian pesan teks dan keakuratan *descriptory* cukup baik, sehingga dapat dimanfaatkan untuk keperluan perlindungan data atau informasi rahasia dalam media gambar. Pada skripsi telah dilakukan implementasi metode F5 untuk menyisipkan pesan ke dalam citra BMP, hasil implementasi kemudian dilakukan pengujian kapasitas pesan yang disisipkan terhadap banyaknya variasi

warna pada *file* citra BMP. Selain itu dilakukan penyisipan pesan dan citra lain (BMP) kedalam *file* citra carier.

Berdasarkan uraian diatas penulis mengangkat judul "**Perancangan Aplikasi Pengamanan Gambar Dengan Metode F5**"

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas adapun yang menjadi rumusan masalah pada penulisan skripsi ini adalah :

1. Bagaimana merancang suatu pengamanan data media gambar kriptografi dan stenografi menggunakan metode F5 ?
2. Bagaimana menerapkan aplikasi media gambar berupa BMP kedalam pengamanan gambar?

## 1.3 Batasan Masalah

Agar pembahasan tidak menyimpang dari tujuannya maka dilakukan pembatasan masalah sebagai berikut:

1. Pemrograman perancangan pengamanan gambar menggunakan aplikasi editor Visual Studio 2010.
2. Aplikasi pengamanan menggunakan kriptografi dan stegnografi berupa media gambar BMP menggunakan metode F5.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Mendapatkan aplikasi pengamanan data gambar metode F5
2. Mampu melakukan pengamanan data teks di dalam media gambar dengan metode F5.
3. Memberikan pengamanan data dalam media gambar dengan format BMP.

### **1.5 Manfaat Penelitian**

Manfaat dari penulisan ini adalah

1. Menambah pengetahuan tentang pengamanan gambar metode F5.
2. Memanfaatkan aplikasi yang dapat melindungi data didalam media gambar berupa BMP.
3. Sebagai bahan referensi bagi peneliti lain yang ingin merancang aplikasi pengamanan gambar dengan metode F5 dan kriptografi.

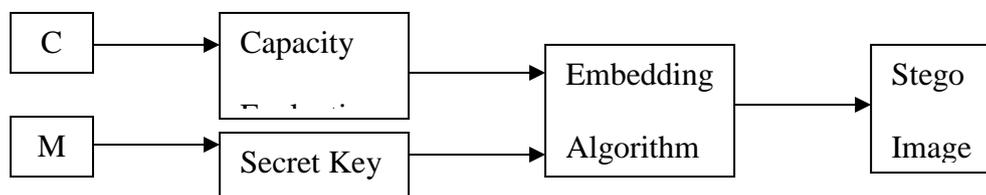
## BAB II

### TINJAUAN PUSTAKA

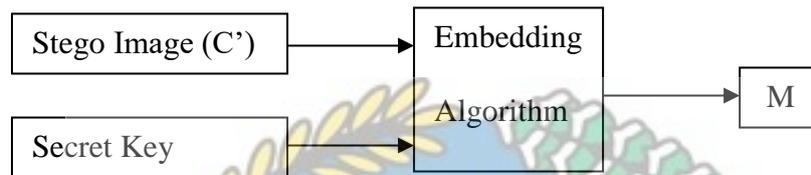
#### 2.1 Steganografi

*Steganografi* berasal dari bahasa Yunani “*steganos*” yang artinya “tersembunyi” atau “terselubung” dan “*graphein*” yang artinya “menulis”. *Steganografi* dapat diartikan “tulisan tersembunyi” (*cover writing*). *Steganografi* adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui (Harjo dkk, 2016).

*Steganografi* membutuhkan dua properti: wadah penampung dan data rahasia yang akan disembunyikan. *Steganografi* digital menggunakan media digital sebagai wadah penampung, misalnya: voice, video, image dan teks. Data rahasia yang disembunyikan juga dapat berupa voice, video, image dan teks. Proses steganografi bisa dilihat pada gambar 1 dan gambar 2 (Harjo dkk, 2016)



**Gambar 2.1** Cara Penyembunyian Pesan Steganografi



**Gambar 2.2 Cara Pengambilan Pesan Steganografi**

Adapun jenis citra yang dapat disisipi pesan dalam steganografi adalah sebagai berikut: (Martsanto dan Jazuli, 2016)

1. *JPG / JPEG (Joint Photographic Experts Assemble)*

*JPG* adalah jenis data yang dikembangkan oleh *Joint Photographic Experts Assemble (JPEG)* yang dijadikan standar untuk para fotografer profesional.

2. *GIF (Graphics Interchange Format)*

*GIF* sama seperti *JPG*, adalah format gambar yang sudah cukup lama digunakan dan salah satu yang umum dipakai di internet. *GIF* adalah kepanjangan dari *Graphics Interchange Format*. *GIF* secara alami adalah gambar dengan 8-bit warna, berarti mereka dibatasi oleh palet sebanyak 256 jenis warna, yang dapat dipilih dari model RGB dan disimpan ke *Color Look Up Tablet (CLUT)*, atau sederhananya "*Color Table*". Mereka itu sejatinya adalah palet warna standar, seperti palet "*Web Safe*". Selain bisa transparansi, *GIF* juga mendukung animasi gambar yang membatasi tiap form nya pada 256 warna standar. Dan karena sifatnya yang tidak pecahpecah, *GIF* bisa digunakan untuk menjaga baris dalam tipografi tetap rapi, dan juga bentuk-bentuk geometri.

3. *PNG (Portable Network Graphic)*

*PNG* adalah kepanjangan dari *Portable Network Graphics*. Dikembangkan sebagai alternatif lain untuk *GIF*, yang menggunakan paten dari *LZW*-algoritma kompresi. *PNG* adalah format gambar yang sangat baik untuk grafis internet, karena mendukung transparansi didalam perambah (*browser*) dan memiliki keindahan tersendiri yang tidak bisa diberikan *GIF* atau bahkan *JPG*. Kelebihan *file PNG* adalah adanya warna transparan dan alpha. Warna *alpha* memungkinkan sebuah gambar transparan, tetapi gambar tersebut masih dapat dilihat mata seperti samar-samar atau bening.

4. *BMP (Bitmap)*

*Bitmap* adalah representasi dari citra grafis yang terdiri dari susunan titik (*pixel*) yang tersimpan di memori komputer. Nilai setiap titik diawali oleh satu *bit* data untuk gambar hitam putih atau lebih untuk gambar berwarna. Kerapatan titik-titik tersebut dinamakan resolusi, yang menunjukkan seberapa tajam gambar ini ditampilkan, ditunjukkan dengan jumlah baris dan kolom contoh 1024×768)

e. *TIFF (Tagged Image Format File)*

*TIFF* merupakan format gambar terbaik dengan pengertian bahwa semua data dan informasi data *RGB*, data *CMYK*, dan lainnya yang berkaitan dengan koreksi atau manipulasi terhadap gambar tersebut tidak hilang. Format *TIFF* biasa digunakan untuk kebutuhan pencetakan dengan kualitas gambar yang sangat tinggi sehingga ukuran berkas untuk format ini biasanya sangat besar, karena dalam *file* ini gambar tidak dikompresi. Format ini mampu menyimpan gambar dengan kualitas hingga 32 *bit*. Format berkas *TIFF* juga

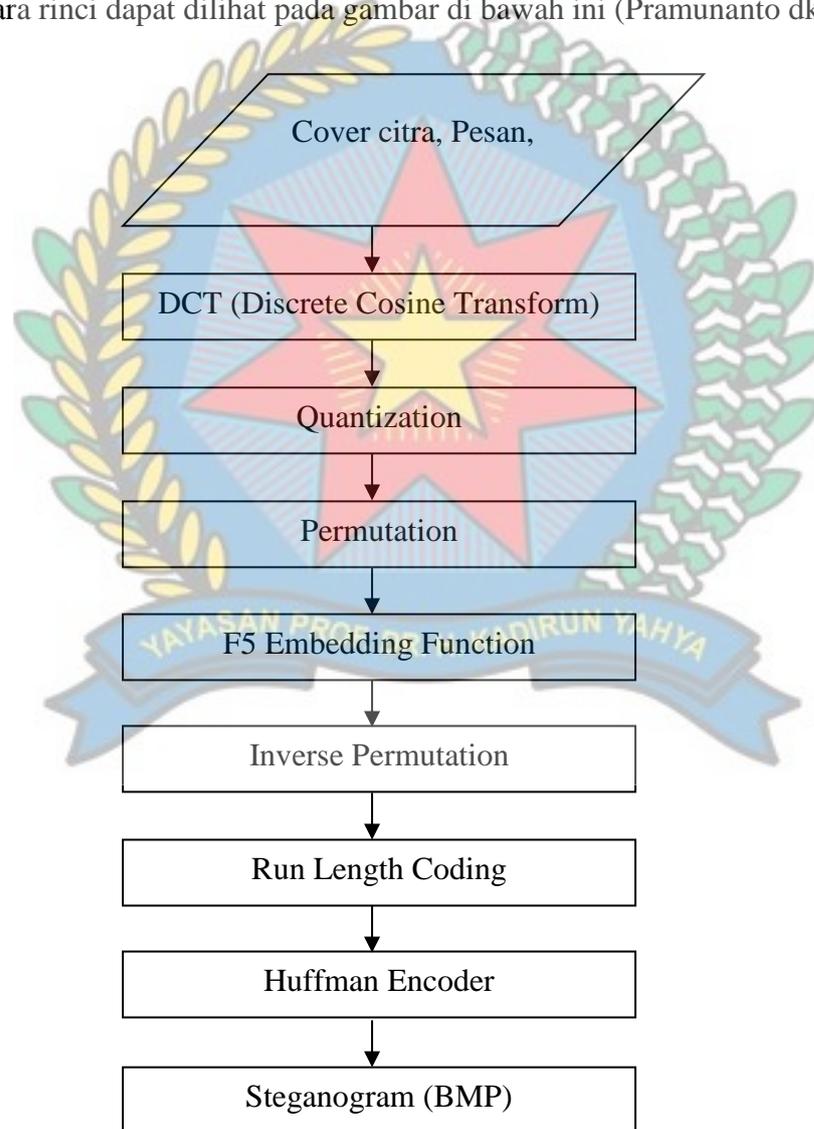
dapat digunakan untuk keperluan pertukaran antar *platform* (*PC, Macintosh, dan Silicom Graphic*). Sebuah pesan steganografi (*plaintext*), biasanya pertama-tama dienkripsikan dengan beberapa arti tradisional, yang menghasilkan *ciphertext*. Kemudian, *coverttext* dimodifikasi dalam beberapa cara sehingga berisi *ciphertext*, yang menghasilkan *stegotext*. Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik *coverttext* lainnya dapat dimanipulasi untuk membawa pesan tersembunyi, hanya penerima yang harus mengetahui teknik yang digunakan dapat membuka pesan dan mendekripsikannya (Martsanto dan Jazuli, 2016).

## 2.2 Enkripsi

Enkripsi adalah proses dimana informasi atau data yang hendak dikirim, diubah menjadi bentuk yang hampir tidak dapat dikenali sebagai informasi pada awalnya dengan menggunakan algoritma tertentu.

Enkripsi data merupakan tahap yang memanfaatkan proses kompresi dari BMP. Proses kompresi akan berhenti sementara pada saat proses kuantisasi selesai. Dari proses tersebut akan didapatkan seluruh koefisien DCT yang telah melalui tahap kuantisasi dan siap untuk tahap enkripsi dengan metode F5. *Input* dari enkripsi data adalah *image*, pesan dan *password*. *Image* sebagai *input*, yang akan dilakukan proses DCT untuk mendapatkan nilai koefisien DCT nya. Tetapi sebelum proses DCT, akan dilakukan konversi gambar dari RGB ke YUV dan pembagian blok piksel dengan ukuran 8x8 (Pramunanto dkk, 2016).

Terdapat beberapa tahap dalam proses penyisipan file pada media image jpeg. Secara rinci dapat dilihat pada gambar di bawah ini (Pramunanto dkk, 2016).



**Gambar 2.3 Enkripsi**

Proses kuantisasi adalah pembagian antara setiap koefisien DCT dengan koefisien dari tabel kuantisasi, dan kemudian dilakukan pembulatan. Setelah tahap kuantisasi selesai, maka akan memasuki tahap inti dari proses enkripsi data. Enkripsi data menggunakan metode F5, pada intinya memanfaatkan nilai nilai

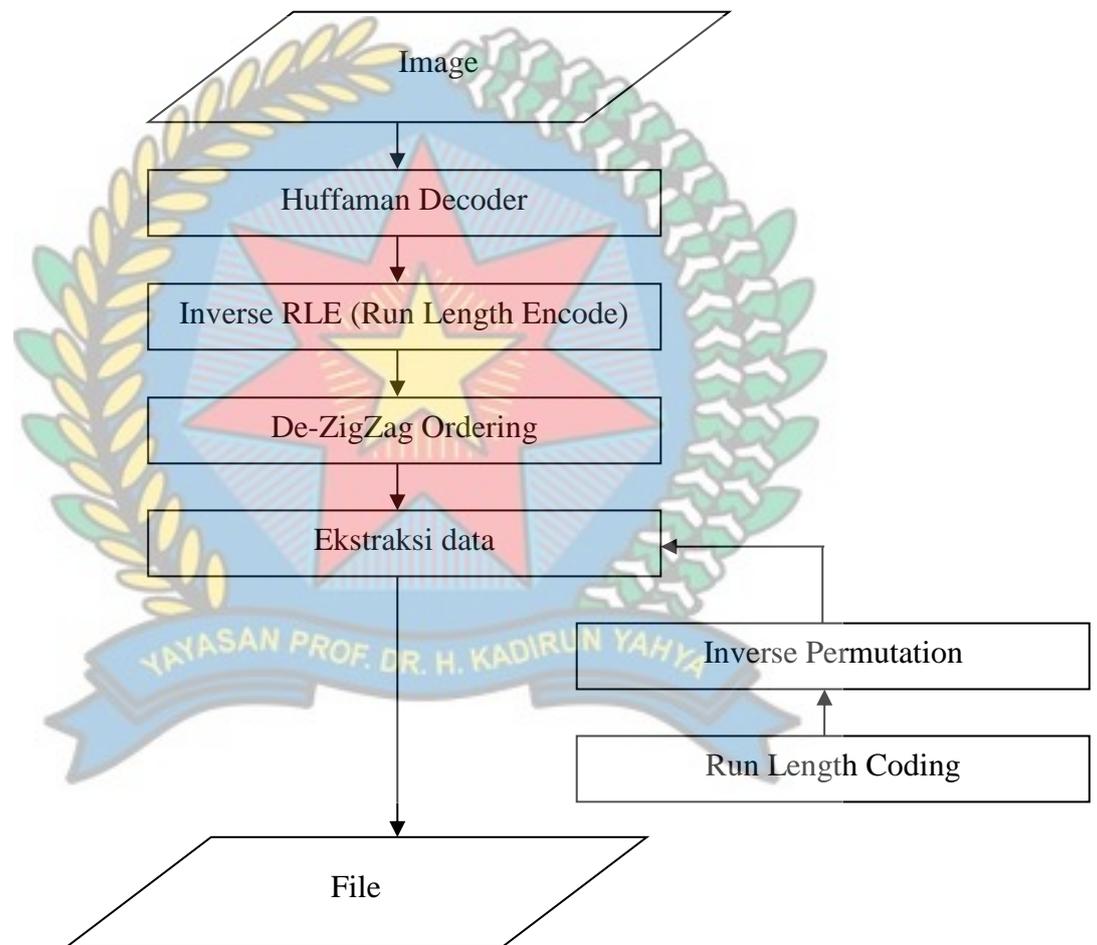
koefisien yang telah dihasilkan dari proses kuantisasi. Sebelum memasuki tahap penyisipan, urutan koefisien akan dilakukan pengacakan dengan metode *permutative straddling* dengan *password* sebagai *input*. *permutative straddling* adalah proses mendapatkan bilangan acak yang dihasilkan oleh *randomgenerator*. Bilangan yang dihasilkan akan digunakan sebagai permutasi yang berfungsi untuk mengacak urutan koefisien.

Informasi penting yang harus diambil dari data yang disisipkan adalah ukuran dari data yang akan disisipkan, terdiri dari 3 *byte* data yang memungkinkan penyisipan sampai ukuran 16.777.215 *byte*. Tahap berikutnya proses penyisipan file dengan metode F5. Semua koefisien DCT akan dicek satu persatu sesuai dengan urutan yang telah diacak oleh permutasi.

Koefisien yang bernilai 0 dan nilai DC tidak dilakukan pemrosesan penyisipan. Proses penyisipan akan lebih rinci dijelaskan pada sub bab *matriks encoding*. *Inverse permutasi* atau pengembalian urutan permutasi ke urutan awal, merupakan tahap setelah proses penyisipan selesai. Koefisien yang telah mengalami perubahan akan dilanjutkan ke proses kompresi BMP selanjutnya, dengan melakukan RLC dan *Huffman code* (Pramunanto dkk, 2016).

### 2.3 Dekripsi

Proses dekripsi merupakan proses untuk menggali informasi yang tersembunyi didalam *image* steganografi. Tahapan proses ekstraksi dapat dilihat lebih rinci pada diagram gambar di bawah ini (Pramunanto dkk, 2016).



**Gambar 2.4 Dekripsi**

Proses dekripsi memerlukan *input image* BMP yang telah terenkripsi dan password. Tahap pertama yang dilakukan adalah mendapatkan koefisien DCT yang berasal dari *image* BMP. Koefisien DCT ini diperoleh ketika proses zigzag ordering selesai (Pramunanto dkk, 2016).

Nilai-nilai koefisien DCT yang didapatkan akan menjadi input dari tahap ekstraksi. Tahap selanjutnya adalah mendapatkan urutan acak koefisien DCT yang

sesuai dengan urutan acak pada saat proses penyisipan, untuk memperoleh urutan yang benar menggunakan *permutative straddling*. Urutan acak yang dihasilkan *permutative straddling* berdasarkan pada *password* yang dimasukkan, apabila *password* yang dimasukkan berbeda, maka tidak akan mendapatkan urutan yang sebenarnya

#### **2.4 Keamanan Data**

Keamanan data merupakan bagian dari perkembangan teknologi informasi. Ketika berpikir bahwa data yang dimiliki merupakan data yang sangat penting, semua berusaha untuk melindunginya agar jangan sampai jatuh ke tangan orang yang tidak bertanggung jawab. Tetapi buat sebagian orang, mereka justru tidak mengetahui sepenting apakah data yang mereka miliki. Karena ketidaktahuan tersebut, mereka baru menyadari bahwa data yang mereka miliki sangat penting setelah mengalami kecurian data dan mengalami kerugian. Data di sini bisa bersifat umum tidak terbatas pada data digital saja, tetapi juga seperti data diri (KTP, ijazah, sertifikat, dan lain-lain). Data yang menyangkut informasi pribadi tidak seharusnya diumbar sembarang seperti pada blog, situs jejaring pertemanan, email, selebaran, fotokopi KTP di buang sembarangan dan lain-lain. (Susilo, 2013:59).

Masalah keamanan merupakan salah satu aspek terpenting dari sebuah sistem informasi. Masalah keamanan sering kurang mendapat perhatian dari para perancang dan pengelola sistem informasi. Masalah keamanan sering berada di urutan setelah tampilan, atau bahkan di urutan terakhir dalam daftar hal-hal yang

dianggap penting. Apabila mengganggu performansi sistem, masalah keamanan sering tidak dipedulikan, bahkan ditiadakan.

Informasi menentukan hampir setiap elemen dari kehidupan manusia. Informasi sangat penting artinya bagi kehidupan karena tanpa informasi maka hampir semuanya tidak dapat dilakukan dengan baik. Contohnya, jika membeli tiket penerbangan dan membayarnya dengan menggunakan kartu kredit, informasi mengenai diri nantinya disimpan dan dikumpulkan serta digunakan oleh bank dan penerbangan. Demikian juga halnya saat membeli obat di apotik. Harus mendapat resep dari dokter dan memberikan resep tersebut ke pelayan apotik. Resep itu merupakan satu informasi yang disampaikan dokter ke pihak apotik tentang obat yang dibutuhkan.

Kemajuan sistem informasi memberikan banyak keuntungan bagi kehidupan manusia. Meski begitu, aspek negatifnya juga banyak, seperti kejahatan komputer yang mencakup pencurian, penipuan, pemerasan, kompetisi, dan banyak lainnya. Jatuhnya informasi ke pihak lain, misalnya lawan bisnis, dapat menimbulkan kerugian bagi pemilik informasi. Sebagai contoh, banyak informasi milik perusahaan yang hanya boleh diketahui oleh orang-orang tertentu di perusahaan tersebut, seperti misalnya informasi tentang produk yang sedang dalam pengembangan. Algoritma dan teknik yang digunakan untuk menghasilkan produk tersebut. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas tertentu.

## 2.5 Metode F5

F5 menggunakan mekanisme *straddling* yang berfungsi mengacak letak semua koefisien DCT terlebih dahulu dengan menggunakan permutasi. F5 kemudian menyisipkan data steganografi menurut urutan permutasi tersebut dan mengirimkannya kepada tahap *Huffman Code* dalam urutan sebenarnya sesudah penyisipan selesai, urutan permutasi itu sendiri didapat dari *password* yang dimasukkan oleh *user* (Pramunanto dkk, 2016).

Dengan *password* yang benar, *user* lain akan mendapatkan urutan permutasinya dan dapat membaca data yang tersembunyi dengan benar. Banyak Metode untuk menghasilkan bilangan random yang berfungsi untuk generate bilangan acak yang digunakan untuk pengacakan index, antara lain yang digunakan adalah *Blum Blum Shub* (BBS). *Blum Blum Shub* (BBS) adalah sebuah *pseudo random number generator* yang dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum, dan Michael Shub. BBS mengambil bentuk sebagai berikut:

$$X_{n+1} = (X_n)^2 \bmod M$$

dimana :

$X_{n+1}$  = bilangan acak ke n+1 dari deretnya

$X_n$  = bilangan acak sebelumnya

M = modulus

### a. Matriks Encoding

*Matrix encoding* adalah penghitungan kode *Hamming* yang sesuai (1, (2k-1), k) dengan menghitung ukuran blok pesan k dari panjang pesan dan jumlah koefisien-koefisien non DC yang tidak nol. Kode Hamming (1, 2k-1, k)

merupakan pengkodean pesan rahasia  $k$ -bit dari  $m$  kata pesan kedalam  $n$  bit kata kode  $a$  dengan  $n=2k-1$ . Kode Hamming dapat merecover dari single bit yang error dalam kode.

Image yang memiliki beberapa variasi warna dengan berbagai ukuran *pixel*. Ukuran *Image* yang diuji antara lain  $8 \times 8 \text{ pixel}$ ,  $32 \times 32 \text{ pixel}$ ,  $64 \times 64 \text{ pixel}$ ,  $128 \times 128 \text{ pixel}$ ,  $256 \times 256 \text{ pixel}$ ,  $512 \times 512 \text{ pixel}$ . *File* yang disisipkan sebesar 1 *Byte*. Pada Tabel 1 ditampilkan hasil kemampuan image dalam kemampuan kapasitas. Semakin besar *image* maka semakin besar pula kapasitas yang dihasilkan, demikian juga dengan keberagaman warna semakin banyak, maka kapasitas yang dihasilkan juga semakin meningkat (Pramunanto dkk, 2016).

Metode F5 merupakan suatu teknik untuk menyisipkan bit pesan kedalam bit koefisien DCT hasil kuantisasi yang telah dipermutasi, penilaian sebuah algoritma steganografi yang baik salah satunya dapat dipandang dari banyaknya pesan yang dapat diterapkan dalam sebuah berkas elektronik. Proses embeddig terdiri dari langkah-langkah berikut (Sianturi dan Hutagaol, 2019):

1. Sediakan pesan yang disisipkan dalam audio
2. Menghitung kapasitas tanpa embedding matriks  $C=hDCT/ 64-h(0)-h(1)+0.49h(1)$ , dimana  $hDCT$  adalah jumlah koefisien DCT  $h(0)$ .
3. Password ditentukan oleh pengguna untuk menghasilkan nilai PSNR.
4. Jika ukuran pesan sesuai dengan perkiraan kapasitas maka proses embed berlaut, lain daripada itu error yang menunjukkan panjang maksimal yang akan ditampilkan

## 2.6 Perancangan

Untuk membuat tampilan yang menarik memang tidak mudah dilakukan. Seorang perancang tampilan selain harus mempunyai jiwa seni yang memadai, ia juga harus mengerti selera pengguna secara umum. Hal lain yang perlu disadari oleh seorang perancang tampilan adalah bahwa ia harus bisa meyakinkan pemrogramnya bahwa apa yang ia bayangkan dapat diwujudkan dengan peranti bantu yang tersedia (Santoso, 2010:20)

Perancangan merupakan proses pengolahan hasil analisis perangkat lunak menjadi rencana pengembangan perangkat lunak dan batasan-batasan perangkat lunak atau masalah yang mungkin dihadapi dalam pengembangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan arsitektur, perancangan modul, dan perancangan antarmuka.

Bagi perancang antarmuka, hal yang sangat penting untuk ia perhatikan adalah mendokumentasikan semua pekerjaan yang dilakukan. Dokumentasi rancangan dapat dikerjakan atau dilakukan dengan beberapa cara :

1. Membuat sketsa pada kertas
2. Menggunakan peranti purwarupa GUI
3. Menuliskan keterangan yang menjelaskan tentang kaitan antara jendela.
4. Menggunakan peranti bantu CASE (*Computer Aided Software Engineering*).

Cara kedua dan keempat tidak selalu dapat diterapkan, karena peranti tersebut biasanya harus dibeli dan seringkali cukup mahal. Cara ini kebanyakan diterapkan pada pembuatan antarmuka grafis untuk suatu jenis pekerjaan berskala besar.

### 2.6.1 Cara Pendekatan

Sebuah program aplikasi pastilah ditujukan kepada pengguna, yang utama, bukan perancangan program aplikasi tersebut. Program aplikasi pada dasarnya dapat dikelompokkan dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula dan program aplikasi yang akan digunakan oleh pengguna umum, yang juga sering dikenal dengan sebutan *public software*. Karena perbedaan pada calon pengguna, maka perancang program antarmuka perlu memperhatikan hal ini (Santoso, 2010:21).

Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi yang serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna maupun ragam antarmuka yang akan digunakan. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan, yakni pendekatan yang disebut dengan pendekatan perancangan berpusat ke pengguna (*user centered design approach*). Cara pendekatan ini berbeda pendekatan perancangan oleh pengguna (*user design approach*).

Pendekatan perancangan berpusat ke pengguna adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna di sini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancangan antarmuka sedang menggambar wajah antarmuka yang mereka inginkan. Dengan kata lain,

perancangan dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya. Sementara perancangan menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna, seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan (Santoso, 2010:22)

Pada perancangan oleh pengguna, pengguna sendirilah yang merancang wajah antarmuka yang diinginkan. Di satu sisi, cara ini akan mempercepat proses pengimplementasian modul antarmuka. Tetapi di sisi yang lain, hal ini justru sangat memberatkan pemrogram karena apa yang diinginkan pengguna belum tentu dapat diimplementasikan dengan mudah, atau bahkan tidak dapat dikerjakan dengan menggunakan peranti bantu yang ada.

Perancang program aplikasi yang dimasukkan dalam kelompok kedua, atau *public software*, perlu menganggap bahwa program aplikasi tersebut akan digunakan oleh pengguna dengan berbagai tingkat kepandaian dan karakteristik yang sangat beragam. Di satu sisi keadaan ini dapat ia gunakan untuk memaksa pengguna menggunakan antarmuka yang ia buat, tetapi pada sisi lain pemaksaan itu akan berakibat bahwa program aplikasinya menjadi tidak banyak penggunanya. Satu kunci penting dalam pembuatan modul antarmuka untuk program-program aplikasi pada kelompok ini adalah dengan melakukan *customization*. Dengan *customization* pengguna dapat menggunakan program aplikasi dengan wajah antarmuka yang sesuai dengan selera masing-masing pengguna.

Salah satu contoh dari adanya kemampuan yang dimiliki oleh sebuah program aplikasi atau sistem operasi yang dapat disesuaikan dengan karakteristik pengguna adalah pengaturan desktop pada OS X versi 10.5 favoritnya, sehingga pengguna dapat mengubahnya sesuai keinginan justru akan membuat mata pengguna itu sakit, dikarenakan mata harus melakukan akomodasi maksimum terus menerus untuk menyesuaikan dengan warna tampilan yang ada.

Selain cara pendekatan yang dijelaskan di atas, Anda yang terbiasa menulis program-program aplikasi mungkin mempunyai cara khusus untuk berhadapan dengan pengguna. Tetapi perlu Anda ingat bahwa apapun cara yang Anda gunakan, Anda tetap harus mempunyai pedoman bahwa pada akhirnya program itu bukan untuk Anda sendiri, tetapi akan digunakan oleh orang lain. Dengan kata lain, jangan pernah mengabaikan pendapat (calon) pengguna program aplikasi Anda (Santoso, 2010:22).

### **2.6.2 Prinsip Dan Petunjuk Perancangan**

Antarmuka pengguna secara alamiah terbagi menjadi empat komponen model pengguna, bahasa perintah, umpan balik, dan penampilan informasi. Model pengguna merupakan dasar dari tiga komponen yang lain.

Model mental pengguna merupakan model konseptual yang dimiliki oleh pengguna ketika ia menggunakan sebuah sistem atau program aplikasi. Model ini memungkinkan seorang pengguna untuk mengembangkan pemahaman mendasar tentang bagian yang dikerjakan oleh program, bahkan oleh pengguna yang sama sekali tidak mengetahui teknologi komputer. Dengan pertolongan model itu

pengguna dapat mengantisipasi pengaruh suatu tindakan yang dilakukan dan dapat memilih strategi yang cocok untuk mengoperasikan program tersebut. Model pengguna dapat berupa suatu simulasi tentang keadaan yang sebenarnya dalam dunia nyata, sehingga ia tidak perlu mengembangkannya sendiri dari awal.

Setelah pengguna mengetahui dan memahami model yang diinginkan, dia memerlukan peranti untuk memanipulasi model itu. Peranti pemanipulasian model ini sering disebut dengan bahasa perintah (*command language*), yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasionalkan (Insap Santoso; 2014:23).

Komponen ketiga adalah umpan balik. Umpan balik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa bentuk umpan balik terutama ditujukan kepada pengguna pengguna yang belum berpengalaman dalam menjalankan program sebuah aplikasi. Umpan balik dapat digunakan untuk member keyakinan bahwa program telah menerima perintah pengguna dan memahami maksud perintah tersebut.

Komponen keempat adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini perancang harus menampilkan pesan-pesan tersebut seefektif mungkin sehingga mudah dipahami oleh pengguna. Setelah

memahami beberapa prinsip dalam perancangan antarmuka pengguna. Pada bagian berikut ini akan diberikan petunjuk singkat tentang perancangan antarmuka yang akan Anda lakukan sebagai seorang perancang tampilan.

### 2.6.3 Urutan Perancangan

Perancangan dialog, seperti halnya perancangan sistem yang lain, harus dikerjakan secara atas ke bawah. Proses perancangannya dapat dikerjakan secara bertahap sampai rancangan yang diinginkan terbentuk, yaitu sebagai berikut

#### a. Pemilihan ragam dialog

Untuk suatu tugas tertentu, pilihlah ragam dialog yang menurut perkiraan cocok untuk tugas tersebut. Ragam dialog dapat dipilih dari sejumlah ragam dialog yang telah dijelaskan pada bab-bab sebelumnya. Pemilihan ragam dialog dipengaruhi oleh karakteristik populasi pengguna, tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut.

#### b. Perancangan Struktur Dialog

Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna sebaiknya banyak dilibatkan, sehingga pengguna langsung mendapatkan umpan balik dari diskusi yang terjadi. Pada tahap ini suatu purwarupa dialog seringkali dibuat untuk memberikan gambaran lebih jelas kepada pengguna.

#### c. Perancangan format pesan

Pada tahap ini tata letak tampilan dan keterangan tekstual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengefektifkan pengguna tombol.

## 2.7 Unified Modeling Language (UML)

*Unified Modelling Language* (UML) Menurut (Haviluddin) 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 yang terkait dengan objek.

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam *fase* sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Saat ini para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep pemodelan *Object Oriented* karena

konsep ini menganalogikan system seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik.

### 2.7.1 Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller J. (2004), "*The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model*". Berikut tujuan utama dalam desain UML adalah (Haviluddin, 2011)

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan *visual* yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

## 2.7.2 Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

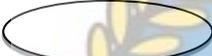
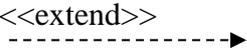
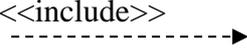
### 1. Use Case Diagram

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Triandini dan Suardika, 2012).

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

| NO. | SIMBOL  | NAMA                  | KETERANGAN  |
|-----|---|-----------------------|---|
| 1.  |    | <i>Use case</i>       | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .   |
| 2.  |    | <i>Actor</i>          | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. |
| 3.  |  | <i>Association</i>    | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.  |
| 4.  |  | <i>Extend</i>         | Relasi <i>use case</i> dimana proses bersangkutan akan dilanjutkan ke proses yang dituju.   |
| 5.  |  | <i>Include</i>        | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>use case</i> tambahan.  |
| 6.  |  | <i>Generalization</i> | Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.   |

Sumber: Rosa dan Shalahuddin (2016)

## 2. Sequence diagram

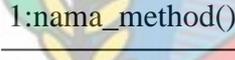
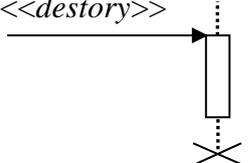
Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case (Triandini dan Suardika, 2012).

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

**Tabel 2.2 Simbol-Simbol Sequence Diagram**

| NO. | SIMBOL  | NAMA            | KETERANGAN   |
|-----|---|-----------------|--|
| 1   | ⋮   | <i>Lifeline</i> | Menyatakan kehidupan suatu objek   |
| 2   | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <u>Nama Objek</u><br/>           :<u>Nama kelas</u> </div> | Objek           | Menyatakan objek yang berinteraksi pesan.  |
| 3   | <div style="border: 1px solid black; width: 20px; height: 20px; margin: 0 auto;"></div>   | Waktu Aktif     | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |

Tabel 2.2 Simbol-Simbol *Sequence Diagram* (Lanjutan)

| NO. | SIMBOL  | NAMA                      | KETERANGAN  |
|-----|---|---------------------------|---|
| 4   |    | Pesan Tipe <i>Creat</i>   | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.   |
| 5   |    | Pesan Tipe <i>Call</i>    | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.  |
| 6   |   | Pesan Tipe <i>Send</i>    | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.  |
| 7   |  | Pesan Tipe <i>Return</i>  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 8   |  | Pesan Tipe <i>Destory</i> | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destory</i> .                     |

Sumber: Rosa dan Shalahuddin (2016)

### 3. Activity diagram

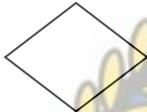
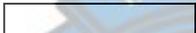
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor. Diagram aktivitas juga banyak digunakan untuk mendefenisikan hal-hal berikut ini (Triandini dan Suardika, 2012):

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefenisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefenisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

**Tabel 2.3 Simbol Activity Diagram**

| NO. | SIMBOL  | NAMA        | KETERANGAN  |
|-----|---|-------------|---|
| 1   |  | Status Awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2   |  | Aktivitas   | Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.                |

Tabel 2.3 Simbol Activity Diagram (Lanjutan)

| NO. | SIMBOL  | NAMA            | KETERANGAN   |
|-----|---|-----------------|--|
| 3   |    | <i>Decision</i> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.                    |
| 4   |    | <i>Join</i>     | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.           |
| 5   |    | Status Akhir    | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6   |  | <i>Swimlane</i> | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.       |

Sumber: Rosa dan Shalahuddin (2016)

UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Berikut beberapa notasi dalam UML diantaranya :

#### 1. *Actor*

*Actor* menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.

## 2. *Class Diagram*

Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.

## 3. *Use Case* dan *Use Case Specification*

*Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, *design*, *testing*, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat *use case* menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem dan tidak menentukan kebutuhan non-fungsional.

## 4. *Realization*

*Realization* menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.

## 5. *Interaction*

*Interaction* digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.

### 2.8 Bahasa Pemrograman **Visual Basic 2010**

Visual Basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi Windows. Visual Basic 2010 atau Visual Basic 9 adalah versi terbaru yang telah diluncurkan oleh Microsoft bersama C#, visual C++, dan Visual Web Developer dalam satu paket Visual Studio 2010.

Visual Basic 2010 merupakan aplikasi pemrograman yang menggunakan teknologi *.NET Framework*. Teknologi *.NET Framework* merupakan komponen Windows yang terintegrasi serta mendukung pembuata, penggunaan aplikasi, dan halaman web. Teknologi *.NET Framework* mempunyai 2 komponen utama, yaitu CLR (*Common Language Runtime*) dan *Class Library*, CLR digunakan untuk menjalankan aplikasi yang berbasis *.NET*, sedangkan *Library* adalah kelas pustaka.

Sebelum menginstall komputer harus memenuhi beberapa persyaratan agar Visual Basic 2010 dapat dijalankan dengan baik. Adapun, persyaratan (*System Requirements*) yang harus dipenuhi.

## BAB III

### METODE PENELITIAN

#### 3.1 Analisis Masalah

Keamanan pada data yang terdapat dalam *file* diperlukan dalam proses penyimpanan yang bersifat rahasia. Hal tersebut disebabkan karena adanya kemungkinan pencurian data dalam *file* tersebut pada proses penyimpanan tersebut. Maka perlu dilakukan analisa sistem untuk memahami informasi-informasi yang didapat dan dikeluarkan oleh sistem itu sendiri. Saat ini sistem komputer yang terpasang makin mudah diakses, oleh karena itu *file* yang memiliki data yang akan diamankan harus diencodekan agar struktur dan data dalam *file* tersebut tidak dapat dibaca maupun dikenali tanpa proses decode terlebih dahulu. Oleh sebab itu keamanan data benar-benar menjadi permasalahan yang sangat penting, sehingga diperlukan sistem tingkat keamanan yang dapat terjamin dan bisa terhindar dari pencurian *file* tersebut.

Dalam menangani pengamanan data gambar di dalam steganografi. Pada proses *encode*, diperlukan proses untuk membaca seluruh *bit* data pada *file* untuk diubah menjadi sebuah blok data yang diperlukan agar algoritma *F5* dapat diimplementasikan untuk melakukan *encode* maupun *decode* pada *file* tersebut, sehingga keamanan dan kerahasiaan data dapat terjaga saat melakukan komunikasi dan pertukaran data tidak dapat di curi oleh pihak yang tidak berkepentingan.

Perpaduan kriptografi dan steganografi teks pada citra digital menjadi lebih aman. Metode yang digunakan adalah mengencode teks terlebih dahulu dengan algoritma. Hasil encode tersebut kemudian dimasukkan ke dalam pixel-pixel tertentu pada citra digital berdasarkan algoritma steganografi F5.

### 3.2 Strategi Pemecahan Masalah

Adapun strategi pemecahan masalah dari sistem *encode* data yang dirancang adalah sebagai berikut :

1. Data yang dibuat didalam sebuah *file* masih sangat rentan terhadap pencurian data atau data itu dapat di salah gunakan oleh orang lain, oleh karena itu perlu dibuat *encode* dalam mengamankan data dengan menggunakan metode F5.
2. Agar data tersebut aman dari pencurian didalam *file* maka di buat teknologi untuk mengamankan data tersebut di dalam *file* dengan teknologi steganografi.

### 3.3 Analisa Kebutuhan *Hardware* Dan *Software*

Untuk meningkatkan sistem keamanan data ini dapat berjalan dengan lancar, maka perangkat lunak dan perangkat keras harus mendukung berjalannya sistem ini dengan baik dalam menggunakan sistem ini.

1. Aspek Perangkat Keras

Adapun kebutuhan perangkat keras untuk menciptakan aplikasi keamanan data tersebut terdiri dari :

- a. *Prosesor Intel Core I3*
  - b. *Harddisk*
  - c. *Memory RAM 2 GB,*
  - d. *Monitor 15 dan Keyboard dan Mouse.*
2. *Aspek Perangkat Lunak (software)*
- Setelah perangkat keras terpenuhi kemudian perangkat lunak yang harus dipenuhi, adapun yang digunakan dalam pembuatan keamanan data yaitu sistem operasi windows 7, aplikasi visual studio 2010.

### 3.4 Penerapan Metode F5

Algoritme F5 untuk penyisiapn pesan rahasia pada citra digital. Proses pertama diawali dengan proses input citra digital melalui galeri kamera atau galeri. Selanjutnya citra yang telah di input akan diproses oleh aplikasi.

Nilai RGB dibaca setelah pengguna memilih citra, kemudian nilai tersebut akan dikonversi menjadi YCbCr yang bertujuan untuk membuang komponen yang tidak penting pada citra sesuai dengan tingkat kepekaan mata manusia, sehingga menghemat ruang pada citra. Mata manusia lebih peka terhadap perubahan warna luminance (Y) daripada warna chrominance (Cb, Cr) sehingga yang digunakan untuk masukkan citra adalah warna Y. Konversi warna RGB menjadi YCbCr menggunakan persamaan (1), (2), (3)

$$Y = 0.299 R + 0.587 G + 0.114 B \quad (1)$$

$$CB = -0.1687 R - 0.3313 G + 0.5 B + 128 \quad (2)$$

$$Cr = 0.5 R - 0.418 G - 0.0813 B + 128 \quad (3)$$

Tahapan ini untuk mengidentifikasi dan menghilangkan komponen pada citra berfrekuensi tinggi yang tidak dapat dideteksi oleh mata manusia namun tidak mengurangi kualitas citra. 64 koefisien DCT ini yang akan digunakan untuk menyisipkan bit-bit pesan rahasia yang dimasukkan oleh pengguna

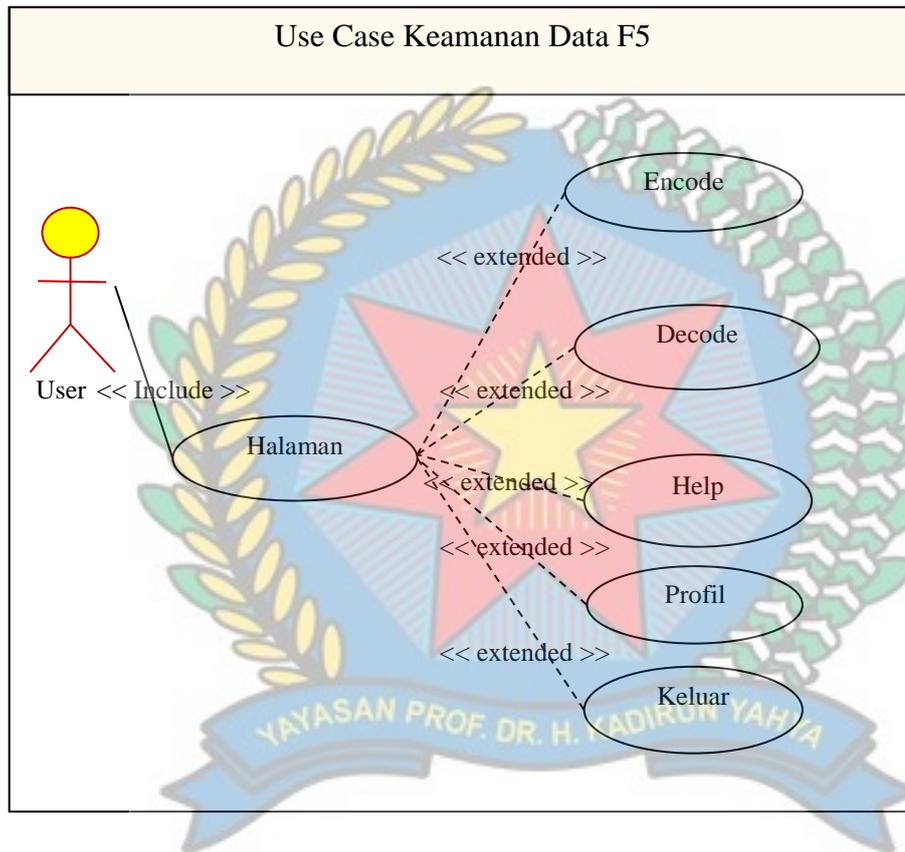
### 3.5 Desain Sistem

Setelah tahapan analisis sistem, maka selanjutnya dibuat suatu rancangan sistem. Perancangan sistem adalah tahapan yang berguna untuk memperbaiki efisiensi kerja suatu sistem yang telah ada. Pada perancangan sistem ini terdiri dari tahap perancangan yaitu :

1. Perancangan *Use Case Diagram*
2. Perancangan *Sequence Diagram*
3. Perancangan *Activity Diagram*
4. Perancangan *Output dan Input*

#### 3.5.1 Use Case Diagram

*Use case* menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsionalitas yang diharapkan dari sebuah sistem *encode file*. Diagram *Use Case* tersebut dapat dilihat pada gambar di bawah ini.



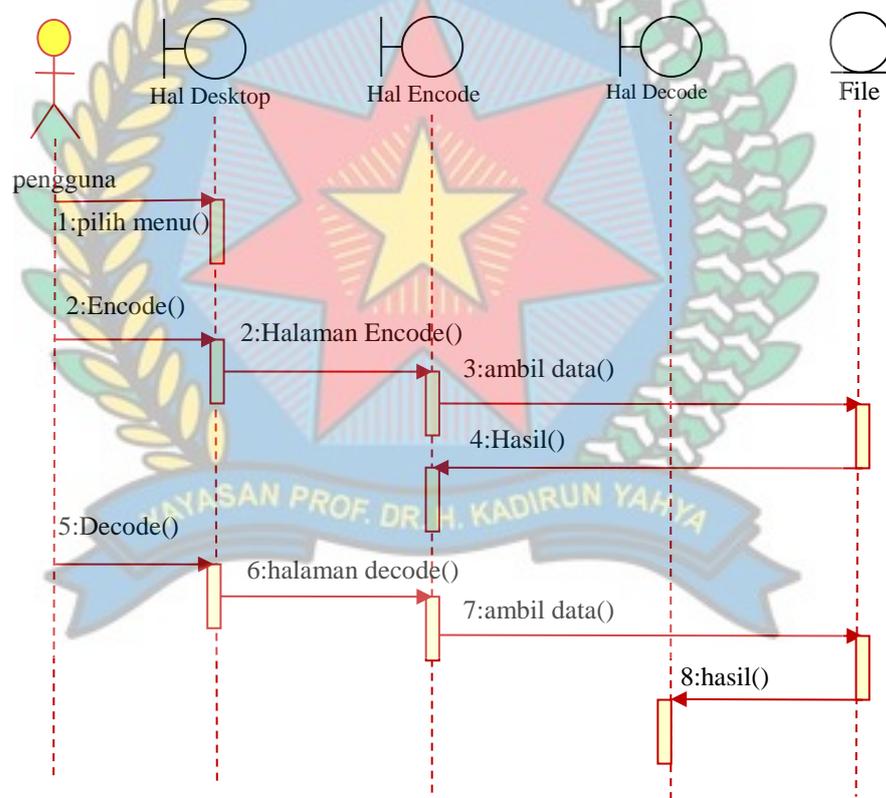
**Gambar 3.1 Diagram Use Case**

### 3.5.2 Sequence Diagram

*Sequence* diagram menunjukkan bagaimana operasi yang dilakukan secara detail. *Sequence* diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang dilakukan oleh seorang *actor* dalam menjalankan sistem, adapun *sequence* yang dilakukan terdiri dari *encode* data dan *decode* data.

### 1. Sequence Halaman Utama

Halaman utama digunakan sebagai pusat semua menu yang terdapat di aplikasi, untuk lebih jelasnya dapat dilihat pada gambar 2.

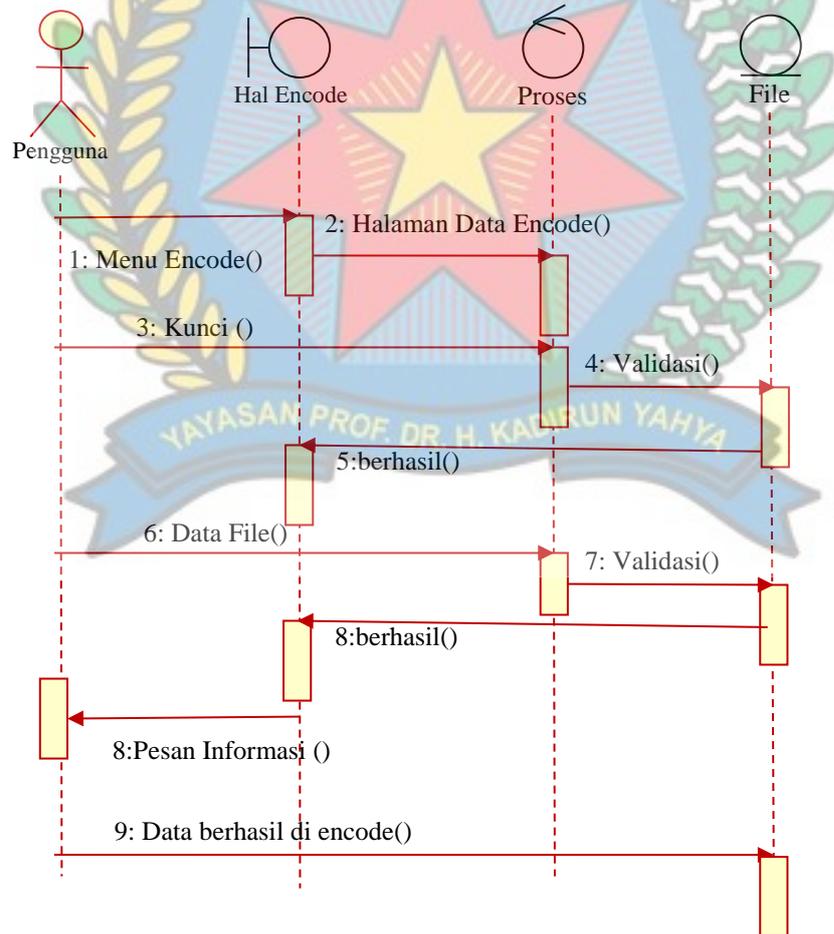


**Gambar 3.2 Sequence Diagram Desktop**

Dari gambar di atas menunjukkan bahwa seorang pengguna dapat melakukan eksekusi dari masing-masing menu yang sudah ditentukan yaitu menu encode untuk mengambil data dalam bentuk *file*, kemudian menu decode untuk mengambil data yang telah di encode.

## 2. Sequence Encode

Encode data digunakan untuk mengubah data asli ke data encode dengan metode hill chipper yang digunakan pada aplikasi tersebut, untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.

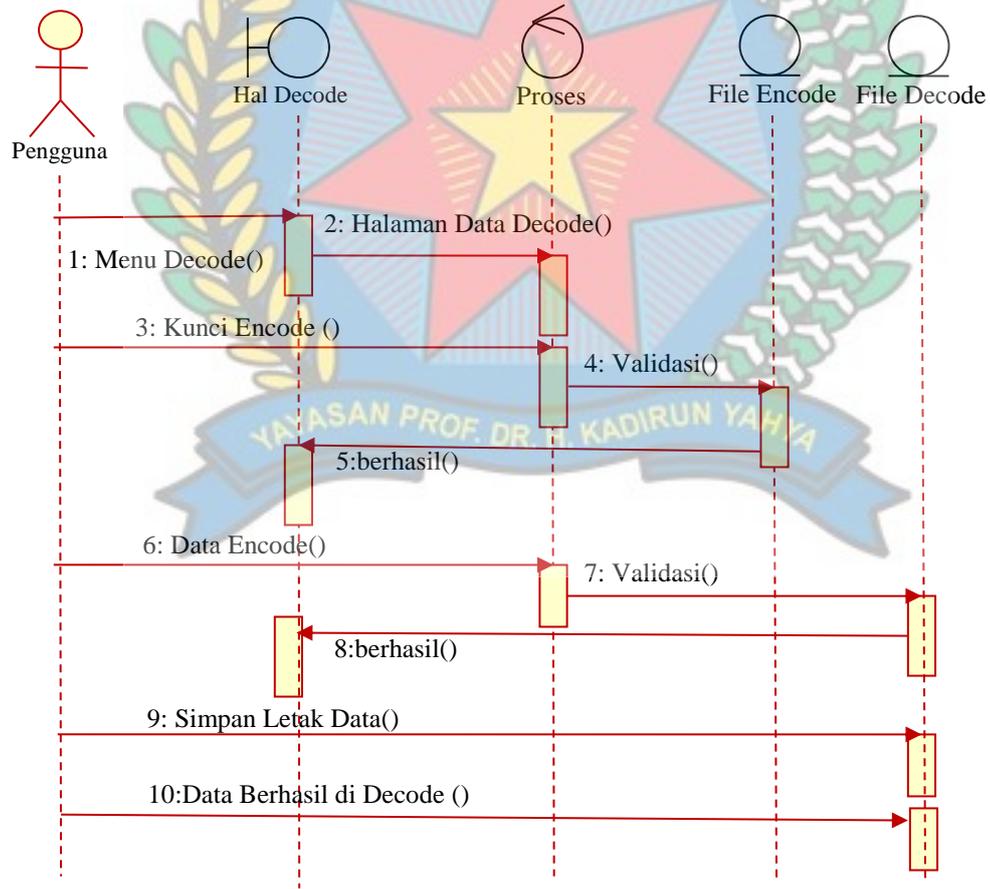


**Gambar 3.3 Sequence Diagram Encode Data**

Dari gambar di atas menunjukkan bahwa seorang pengguna jika ingin melakukan *encode* sebuah data harus terlebih dahulu memasukan gambar dan data yang disembunyikan di dalam media gambar tersebut, mengambil data yang berbentuk *file*, dan melakukan proses *encode*

### 3. Sequence Decode

Decode data digunakan untuk mengubah data encode kemudian data tersebut di ubah ke data yang asli dengan menggunakan tombol decode. untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



s

**Gambar 3.4 Sequence Diagram Decode Data**

Dari gambar di atas menunjukkan bahwa seorang pengguna melakukan perubahan data yang diencode menjadi data yang semula atau data asli dengan proses decode.

#### 4. Sequence Help

Help digunakan untuk memberikan petunjuk cara mempergunakan program ini. untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



**Gambar 3.5 Sequence Diagram Decode Data**

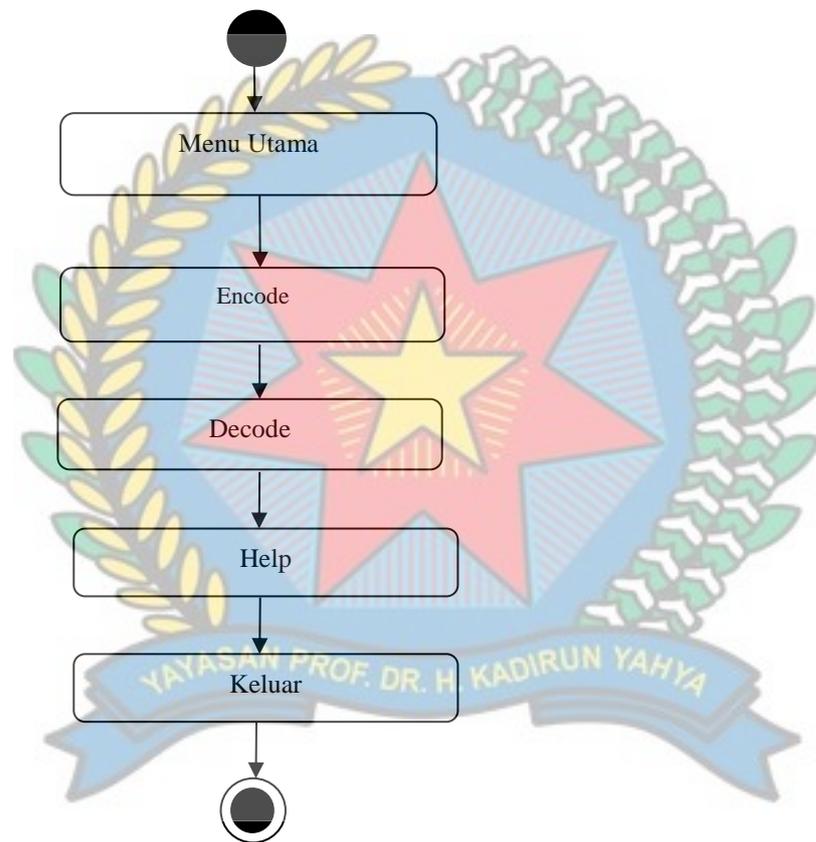
Dari gambar di atas menunjukkan bahwa seorang pengguna jika bingung dapat memanfaatkan menu help.

#### 3.5.3 Activity Diagram

*Activity diagram* ini akan menjelaskan setiap kegiatan yang akan dilakukan pengguna pada sistem nantinya. Dengan menggambarkan setiap aktivitas dari sistem diharapkan sistem yang akan dibangun lebih mudah dipahami.

##### 1. Activity Diagram Menu Utama

*Activity diagram* untuk menu utama. *Activity diagram* dapat dilihat pada gambar di bawah ini.

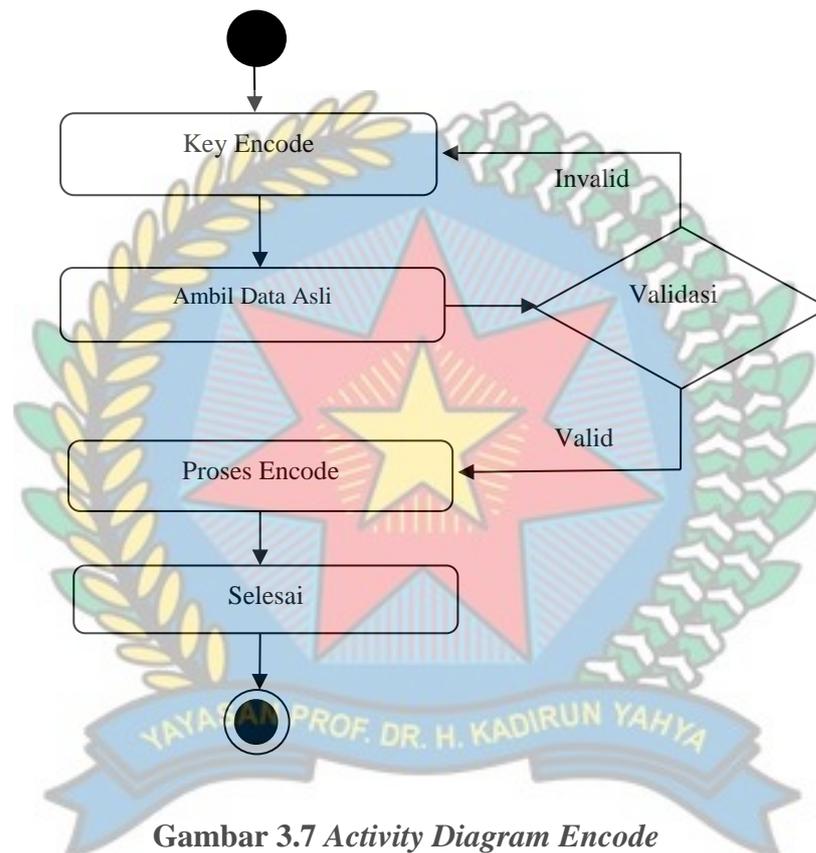


**Gambar 3.6** *Sequence Diagram* Menu Utama

Dari gambar di atas menunjukkan bahwa menu utama ini adalah pusat seluruh program yang ada pada aplikasi tersebut.

## **2. Activity Diagram Encode**

*Activity* diagram untuk proses *encode* data. *Activity* diagram *encode* data dapat dilihat pada gambar di bawah ini.

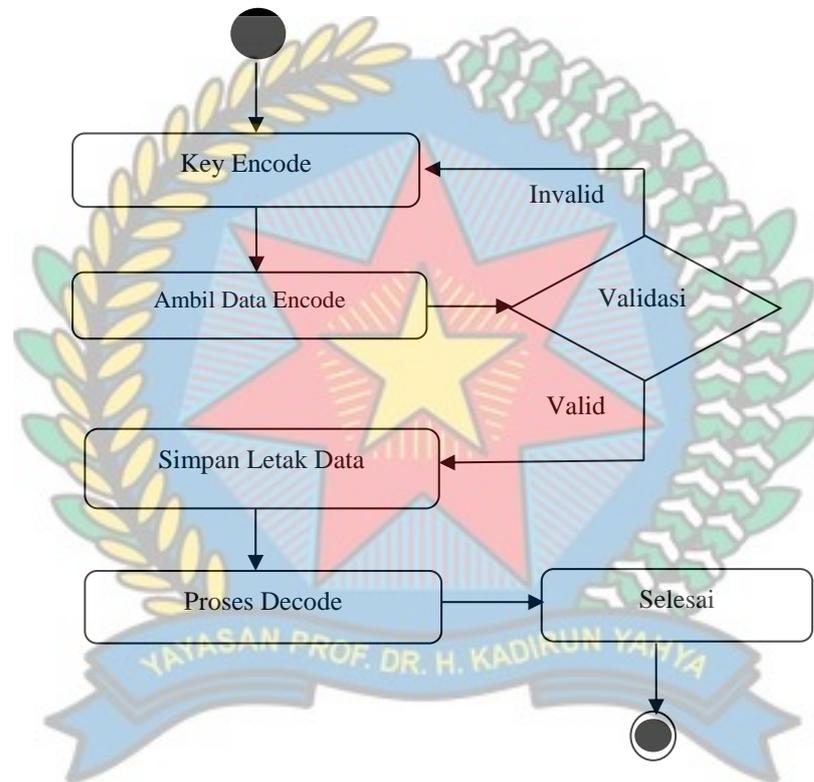


**Gambar 3.7 Activity Diagram Encode**

Pada gambar di atas menjelaskan bahwa pengguna memasukkan kunci panjang 8 *Bit*, lalu memasukkan data *file* yang ingin di encodekan, jika semuanya sudah diinputkan maka proses *encode* selesai dilakukan, jika salah satu belum diinputkan maka kembali diminta untuk menginputkan keseluruhan.

### **3. Activity Diagram Decode**

*Activity* diagram merupakan *activity* diagram untuk proses *decode*. *Activity* diagram tersebut ditunjukkan pada gambar di bawah ini.

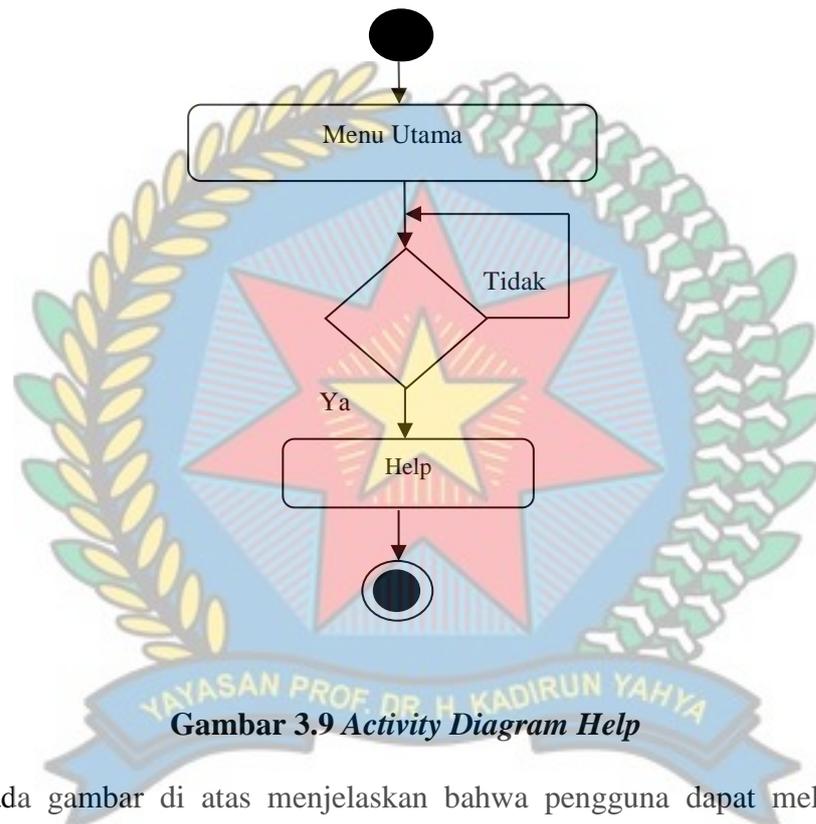


**Gambar 3.8 Activity Diagram Deskripsi**

Pada gambar 8 menjelaskan bahwa pengguna menginputkan sebuah *file* yang telah di encodekan kemudian memberikan kunci encodenya, jika semuanya sudah diinputkan maka proses decode dapat dilakukan, jika salah satu belum diinputkan maka kembali diminta untuk menginputkan keseluruhan.

#### **4. Activity Diagram Help**

*Activity* diagram ini merupakan *activity* diagram untuk proses memberikan petunjuk, dapat dilihat pada gambar di bawah ini.



**Gambar 3.9 Activity Diagram Help**

Pada gambar di atas menjelaskan bahwa pengguna dapat melihat cara penggunaan program ini di menu help.

### 3.6 Rancangan Program

Dalam merancang suatu sistem perlu diketahui hal yang akan menunjang sistem, agar dapat mempermudah pengolahan data nantinya. Pengolahan data ini diharapkan dapat mempermudah dalam hal penyajian, pelayanan dan pembuatan berbagai laporan data yang dibutuhkan.

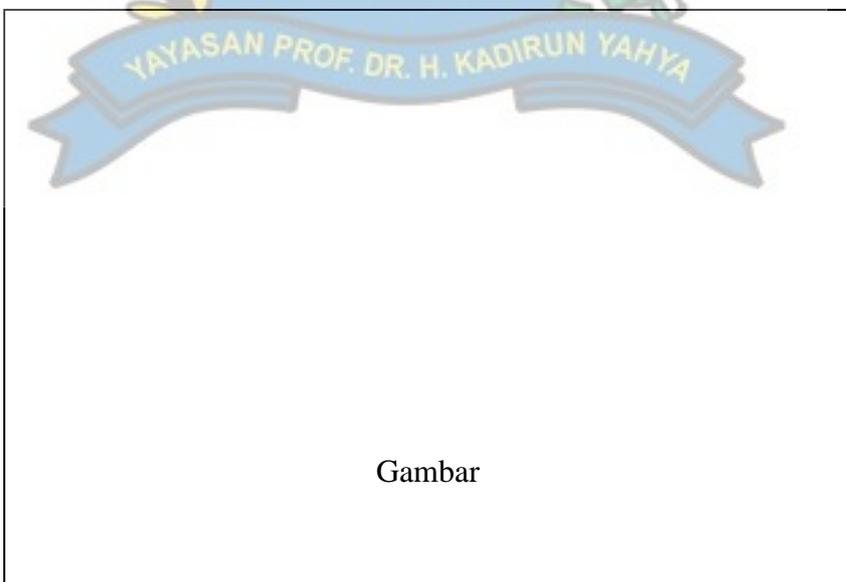
#### 3.6.1 Rancangan *Output*

Rancangan *output* sistem global sebagaimana telah dijelaskan di atas tidak dapat menggambarkan secara keseluruhan proses yang terjadi dalam sistem, sehingga dibutuhkan disain sistem secara detail yang dapat menjelaskan alur

proses yang terjadi di dalam sistem tersebut. Adapun rancangan sistem secara detail yang diusulkan akan dijelaskan satu persatu berikut ini.

### 1. Rancangan *Form* Menu Utama

Dalam perancangan menu utama ini digunakan sebagai pusat dari program perancangan keamanan data dari masing-masing menu, untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.

|   |        |      |        |        |
|---|--------|------|--------|--------|
| Menu Utama  |        |      |        |        |
| Encode  | Decode | Help | Profil | Keluar |
|  <p>Gambar</p> |        |      |        |        |

**Gambar 3.10 Rancangan Form Menu Utama**

Rancangan menu utama atau menu utama terdiri dari beberapa menu yaitu, menu encode, menu decode, menu help, menu profil dan menu keluar. Jika pengguna ingin melakukan encode data klik pada menu encode, jika pengguna ingin mengembalikan ke data asli klik menu decode, Jika pengguna bingung

dalam menjalankan aplikasi ini, pengguna klik menu petunjuk, jika ingin keluar klik menu keluar.

## 2. Rancangan *Form Encode*

Dalam perancangan *form encode* ini berfungsi melakukan pengambilan data untuk melakukan keamanan data tersebut, untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.

| ENCODE             |                                       |  |  |
|--------------------|---------------------------------------|--|--|
| Input Plainteks    | <input type="text"/>                  |  |  |
| File Gambar        | <input type="text"/>                  | <input type="button" value="Cari"/>      | <input type="button" value="Pengamanan Gambar"/> |
| Tampil Gambar      | <input type="text"/>                  | Sumber                                   | <input type="text"/>                             |
|                    |                                       | Tujuan                                   | <input type="text"/>                             |
|                    | <input type="button" value="Proses"/> |  |  |
| Hasil Encode       | <input type="text"/>                  | <input type="button" value="Proses"/>    |  |
| Hasil Steganografi | <input type="text"/>                  | <input type="button" value="Prosesbar"/> |  |
|                    | <input type="button" value="Simpan"/> |  |  |

**Gambar 3.11 Rancangan *Form Encode***

Rancangan form encode yang terdapat pada gambar di atas terdiri dari tombol data, proses, keluar. Berikutnya pengguna diminta masukkan kunci, data dan terakhir adalah hasil encode.

### 3. Rancangan *Form Deskripsi*

Bentuk daripada *form deskripsi* yang dirancang dapat dilihat pada gambar di bawah ini.



| FORM DECODE |                      |
|-------------|----------------------|
| Cari Gambar | <input type="text"/> |
|             | <input type="text"/> |
|             | Proses Pengamanan    |
| Buka Stegno | <input type="text"/> |
|             | <input type="text"/> |
|             | Proses               |
| Pesan       | <input type="text"/> |
| F5          | <input type="text"/> |

**Gambar 3.12 Rancangan *Form Deskripsi***

Rancangan form decode yang terdapat pada gambar di atas terdiri dari tombol data encode, simpan, decode, batal, keluar. Berikutnya pengguna diminta masukkan gambar encode terlebih dahulu, kemudian klik proses.

### 3.6.2 Rancangan *Input*

Sistem ini mempunyai beberapa halaman yang akan menjadi *input*. Dalam perancangannya, sistem yang diusulkan adalah sebagai berikut :

Rancangan *form input* data encode ini digunakan untuk mengambil data dari sebuah *root direktory* yang terletak didalam sebuah *folder* tertentu. Bentuk daripada *form input* data encode yang dirancang dapat dilihat pada gambar di bawah ini.

| Open File                  |   |
|----------------------------|---|
| Tampilan<br>Root Direktori |   |
| File Name                  | <input type="text"/>  |
|                            | *.bmp <input type="button" value="v"/>                                    |
|                            | <input type="button" value="Open"/> <input type="button" value="Cancel"/> |

**Gambar 3.13 Rancangan *Form Input File***

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Tampilan Hasil

Berikut adalah tampilan hasil dan pembahasan dari Aplikasi keamana gambar stegnografi Metode F5.

##### 4.1.1 Tampilan Menu Utama

Tampilan menu utama terdiri dari beberapa tombol yaitu tombol enkripsi, deskripsi, help, profil dan keluar, untuk lebih jelasnya dapat dilihat pada Gambar di bawah ini.

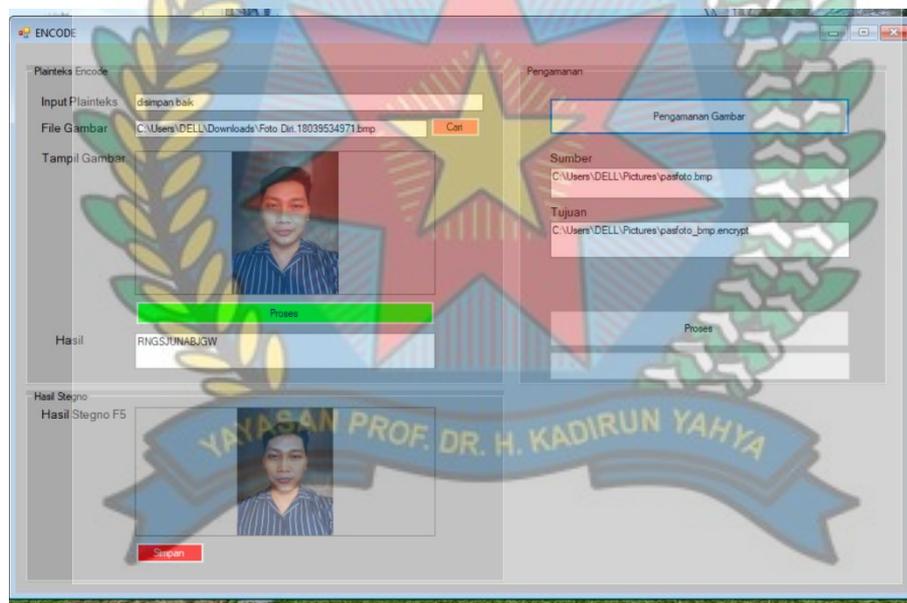


**Gambar 4.1 Tampilan Menu Utama**

Pada Gambar di atas menampilkan menu utama yang berfungsi sebagai pusat seluruh program yaitu enkripsi, dekripsi, help, profil dan keluar.

#### 4.1.2 Tampilan *Encode*

Tampilan *form enkripsi* diminta pengguna untuk memasukan kunci atau *key*, dan memasukan data yang ingin di amankan, untuk lebih jelasnya dapat dilihat pada Gambar di bawah ini.

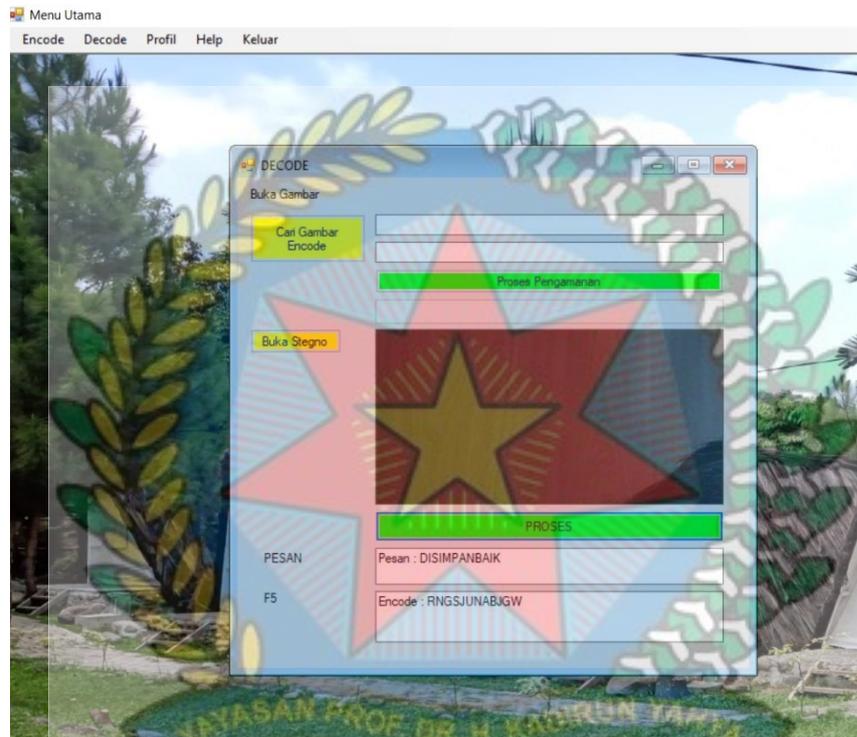


**Gambar 4.2 Tampilan *Form Encode***

Halaman pada gambar di atas menjelaskan bahwa user dapat melakukan memasukan kalimat pesan dan memasukan gambar yang berjenis BMP, selanjutnya tombol proses untuk melakukan esekusi terhadap inputan yang dilakukan user yang menghasilkan enkripsi dan stegnografi. Setelah berhasil dilakukan data tersebut disimpan dengan melakukan pada tombol simpan.

#### 4.1.3 Tampilan *Form Decode*

Tampilan *form dekripsi* diminta user memasukan gambar enkripsi, untuk lebih jelasnya dapat dilihat pada Gambar di bawah ini.



**Gambar 4.3 Tampilan Form Decode**

Halaman gambar di atas menjelaskan halaman dekripsi untuk melakukan eksekusi proses terhadap gambar yang dimasukkan, jika semua berhasil maka menghasilkan pesan asli dan pesan yang dimasukkan kedalam gambar.

#### **4.1.4 Tampilan Form Help**

Tampilan halaman form help memberitahukan informasi tentang cara penggunaan aplikasi keamanan data, untuk lebih jelasnya dapat dilihat pada Gambar di bawah ini.



Gambar 4.4 Tampilan Form Help

#### 4.1.5 Tampilan Form About

Tampilan halaman *form about* memberitahukan informasi tentang identitas programmer, untuk lebih jelasnya dapat dilihat pada Gambar di bawah ini.



Gambar 4.5 Tampilan Form About

## 4.2 Pengujian

Dalam pengujian ini penulis melakukan pengujian dengan hasil aplikasi keamanan data kriptografi dan stegno dengan metode F5, dengan memunculkan data berupa *file* yang ingin di lakukan keamanan data tersebut.

### 4.2.1 Rencana Pengujian

Pada tahap implementasi dan pengujian terhadap keamanan data yang dirancang secara sederhana, agar *user* dapat dengan mudah melakukan keamanan data dengan cepat dan akurat.

**Tabel 4.1 Skenario Pengujian Sistem**

| Komponen yang di uji                 | Pengujian                      | Tingkat pengujian | Jenis pengujian |
|--------------------------------------|--------------------------------|-------------------|-----------------|
| Pengujian pengisian data berupa file | pengisian data user (pengguna) | Sistem            | Blackbox        |
| Pengujian Enkripsi                   | Pengecekan algoritma F5        | Sistem            | Blackbox        |

**Tabel 4.2 Pengujian Sistem data Keamanan Data**

| Kasus hasil uji (Data normal) |                  |                                      |                                      |                             |
|-------------------------------|------------------|--------------------------------------|--------------------------------------|-----------------------------|
| No                            | Data masukan     | Yang diharapkan                      | Pengamatan                           | Kesimpulan                  |
| 1.                            | Data berupa file | Data berupa file yang akan dimasukan | Data berupa file yang di enkripsikan | [✓] diterima<br>[ ] ditolak |

|                                     |                             |  |   |  |
|-------------------------------------|-----------------------------|--|---|--|
|                                     |                             |  | dengan metode des                               |  |
| 2.                                  | Enkripsi file               | Data file yang akan di enkripsi              | Data file berubah menjadi data terenkripsi      | <input checked="" type="checkbox"/> diterima<br><input type="checkbox"/> ditolak |
| <b>Kasus hasil uji (Data salah)</b> |                             |  |   |  |
| <b>No</b>                           | <b>Data masukkan</b>        | <b>Yang diharapkan</b>                       | <b>Pengamatan</b>                               | <b>Kesimpulan</b>  |
| 3.                                  | Masukkan data tidak lengkap | Ada pesan bahwa pengisian data tidak lengkap | Muncul pesan bahwa pengisian data tidak lengkap | <input checked="" type="checkbox"/> diterima<br><input type="checkbox"/> ditolak |

#### 4.2.2 Pengujian Kasus dan Hasil

Dalam pengujian keamanan data dengan metode F5 yang penulis lakukan dengan menggunakan aplikasi gambar yang terdapat pada sistem operasi windows

7. Untuk lebih jelasnya dapat dilihat pada tabel berikut ini.

**Tabel 3. Pengujian Keamanan Data Metode F5**

| No | File Asli | File Enkripsi | Nama File       | Ukuran |
|----|-----------|---------------|-----------------|--------|
| 1. | nando     | CMGUXSYI      | Gambar1.b<br>mp | 250 MB |
| 2. | F5        | VXOFLWFKLQ    | Gambar2.b<br>mp | 300 MB |

Pada tabel di atas dalam pengujian keamanan data metode F5 menggunakan *notepad* dengan kapasitas 250 MB. Adapun hasil dari aplikasi tersebut berbeda hasil karena tergantung biner dan isi dari *file* asli tersebut.

#### 4.3 Pembahasan

Hasil aplikasi sistem keamanan data untuk memberikan kemudahan mengenai pengamanan *stegno* agar sistem keamanan data ini dapat berjalan dengan sempurna, pertama sekali harus ada *file* gambar yang ingin di enkripsikan selanjutnya jalankan aplikasi yang penulis rancang.

#### 4.4 Kelebihan Dan Kekurangan Sistem Yang Dirancang

Sistem yang dirancang mempunyai beberapa kelebihan dan kekurangan ketika diterapkan diantaranya :

1. Kelebihan dari sistem yang dirancang :
  - a. Aplikasi sistem yang dirancang mempercepat proses keamanan data yang terdiri dari aplikasi yang berextension bmp.
  - b. Mempermudah *user* dalam pengolahan data.
  - c. *File* yang sudah di *enkripsi* sangat susah untuk diketahui oleh pihak-pihak yang tidak bertanggung jawab.
2. Kekurangan dari sistem yang dirancang :
  - a. Hanya satu *file* saja yang bisa di encode.
  - b. Tidak dapat mengenkripsi *file* berupa gambar

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari uraian secara teoritis pada sistem keamanan data stegno metode lsb+1 dan krypto metode hill chiper, maka penulis akan mencoba menarik kesimpulan dan akan memberikan saran-saran sebagai berikut:

1. Keamanan gambar yang di enkripsi adalah berupa *file* gambar yang disisipkan teks.
2. Keamanan data dilengkapi dengan dekripsi.
3. Metode yang digunakan Hill Chiper dan LSB+1
4. Bahasa pemrograman keamanan data visual studio 2010.

#### 5.2 Saran

Untuk menyempurnakan sistem yang telah dibuat, penulis memberikan saran-saran sebagai berikut:

1. Perlu adanya pengembangan dari aplikasi ini supaya tidak hanya satu data saja yang dapat di *enkripsi*.
2. Perlu adanya pengembangan agar aplikasi dapat melakukan enkripsi *file image*.
3. Perlu dibuat perbandingan keamanan data dengan metode yang lain seperti 3DES.

## DAFTAR PUSTAKA

- Akbar, M Barkah., Haryanto, Edi Victor. (2012). Aplikasi Steganografi dengan Menggunakan Metode F5. *Jurnal Sistem Informasi Dan Teknologi Informasi*, Vol 4 No 2 Oktober 2015, Hal 165-175. ISSN : 2252-6102.
- Harjo, Teguh Budi., Kapriati, Marly., Susanto, Dwi Ardian. (2016). Aplikasi Steganografi Menggunakan LSB (Least Significant Bit) dan Enkripsi Triple DES Menggunakan Bahasa Pemrograman C#. *Jurnal SISFOTEK Global*, Vol 6 No 1 Maret 2016, Hal 13-17. ISSN : 2088 – 1762.
- Haviluddin. (2011). Memahami Penggunaan UML (Unified Modelling Language). *Jurnal Informatika Mulawarman*, Vol 6 No 1 Februari 2011, Hal 1-15.
- Martsanto, Sandy., Jazuli, Wildan. (2016). Teknik Steganografi Dan Enkripsi Dokumen Guna Menjamin Keamanan Dan Integritas Informasi Dalam Lingkup Organisasi (Studi Kasus Pada PT Saptawara Teknologi Indonesia). *Jurnal Ilmiah Ilmu Komputer*, Vol 2 No 1 April 2016, Hal 19-25. ISSN : 2503-3832.
- Pramunanto, Eko., Muhtadin., Putra, Yanu Perwira Adi., Zaini, Ahmad. (2012). Penulisan Pesan Tersembunyi Pada Citra JPEG dengan Metode F5. *JAVA Journal of Electrical and Electronics Engineering*, Vol 10 No 2 Oktober 2012, Hal 1-8.
- Rosa, A. S., Salahuddin, M., (2016). Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung : Informatika.
- Santoso, Insap. (2010). Interaksi Manusia dan Komputer. Yogyakarta : ANDI.
- Sianturi, Tri Nusanti., Hutagaol, Rinaldo Gomgom. (2019). Penyisipan Pesan Rahasia Kedalam Audio Menggunakan Algoritma F5. *Prosiding Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*.
- Susilo, Andik. (2010). Teknik Cepat Memahami Keamanan Komputer dan Internet. Jakarta : PT Elex Media Komputindo.
- Triandini, Evi., Suardika, I Gede. (2012). Step By Step Desain Proyek Menggunakan UML. Yogyakarta : ANDI.