BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan cara berpikir manusia membuat masyarakat menyadari bahwa teknologi informasi merupakan salah satu *tool* penting dalam peradaban manusia untuk mengatasi (sebagian) masalah derasnya arus informasi. Teknologi informasi (dan komunikasi) saat ini adalah bagian penting dalam manajemen informasi.Selain memiliki potensi dalam mem*filter* data dan mengolah menjadi informasi, teknologi informasi mampu menyimpan dengan jumlah kapasitas jauh lebih banyak dari cara-cara manual. Salah satu pekerjaan manusia yang akan sangat terbantu dengan hadirnya teknologi informasi, dengan keuntungan yang ditawarkan, yaitu pekerjaan manusia dalam menyembunyikan data atau pesan.

Kerahasiaan dan keamanan suatu data pada zaman globalisasi sekarang ini semakin menjadi sebuah kebutuhan vital dalam berbagai aspek kehidupan. Suatu data akan memiliki nilai lebih tinggi apabila menyangkut aspek-aspek keputusan bisnis, keamanan, ataupun kepentingan umum. Dimana data-data tersebut tentunya akan banyak diminati oleh berbagai pihak yang juga memiliki kepentingan di dalamnya. Dengan berkembangnya teknologi informasi semakin berkembang pula kejahatan yang berhubungan dengan data itu sendiri.Dengan berbagai teknik banyak yang mencoba untuk mengakses data yang bukan haknya. Keamanan pada suatu data pada saat ini dapat dibagi menjadi dua, yakni :*Cryptography* dan *Steganography*. Dari dua metode tersebut, metode yang satu

dapat menjaditambahan bagi metode yang lain. Menurut terminologinya *cryptography* adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. Berbeda dengan *steganography*, *steganography* adalah seni menyamarkan / menyembunyikan pesan secara tertulis kedalam pesan lainnya.(Ariyus, 2008).

Teknik *Steganography* ini mempunyai beberapa metode yang digunakan untuk mengamankan suatu data atau pesan salah satunya adalah metode *spread spectrum*. Metode *Spread Spectrum* mentransmisikan sebuah sinyal pita informasi yang sempit kedalam sebuah kanal pita lebar dengan penyebaran frekuensi (Dony Ariyus, 2008).*Spread spectrum image steganography* adalah metode yang menempatkan informasi di dalam derau semua keseluruhan *cover image*. Dengan kata lain pada metode ini pesan yang hendak di sembunyikan dimasukkan secara menyeluruh pada *cover image*, sehingga akan lebih sulit untuk dideteksi keberadaannya dan pesan akan terlihat seperti *noise*. Dalam *steganography* memerlukan suatu media sebagai tempat penyembunyian informasi. Secara teori penyisipan informasi pada data digital dengan menggunakan teknik *steganography* dapat dilakukan pada semua format data digital yang ada dalam komputer sebagai media *cover*nya seperti format teks, format gambar, bahkan untuk format audio dan sebagainya asalkan file-file tersebut mempunyai bit-bit data yang dapat dimodifikasi.

Penggunaan file *image* sebagai salah satu media *steganography* merupakan langkah yang baik. Lalu lintas pertukaran file image di internet merupakan hal biasa, sehingga *steganography* menggunakan file *image* adalah

teknik yang baik untuk mengamankan pesan rahasia melalui media internet. Selain itu jika kita tidak bicara dalam konteks internet, *steganography* juga menjadi media yang paling digemari karena paling sering digunakan sebagai sarana hiburan. Semakin sering file itu atau semakin terlihat file itu maka akan semakin kecil kecurigaan bahwa terdapat pesan tersembunyi dalam file tersebut (Khasadika Donovan, Ekojono, Imam Fahrur Rozi, 2015).

Berdasarkan uraian dan permasalahan diatas, penulis tertarik untuk merancang aplikasi *steganography* yang dapat menyisipkan pesan file berformat PDF ke dalam file *image* sehingga pesan tersebut dapat terjaga kerahasiaannya.

B. Rumusan Masalah

Berdasarkan uraian dari latar belakang diatas, peneliti dapat mengambil rumusan masalah yang akan dilakukan yaitu Bagaimana Enkripsi File PDF to *Image* dengan Metode *Spread Spectrum*?

C. Tujuan Penelitian

Adapun tujuan dari pembu

atan sistem ini untuk merancang sebuah aplikasi *steganography*pada file pdf*to image* dengan metode *spread spectrum*.

D. Batasan Masalah

Untuk dapat menghasilkan penelitian yang lengkap, akurat dan tepat waktu, maka diperlukan adanya ruang lingkup dan batasan masalah, oleh karenanya judul yang diangkat penulis membatasi beberapa masalah.

1. Aplikasi *steganography* ini tidak bisa digunakan untuk format lain (.txt, .doc, dll).

2. Teknik *steganography* yang digunakan adalah metode *spread Spectrum image steganography*.

E. Manfaat Penelitian

Manfaat dari aplikasi *steganography* pdf *to image* dengan metode *Spread Spectrum*yaitu :

a. Manfaat bagi Penulis

   Dapat menerapkan dan mengembangkan ilmu keamanan data dari sebuah citra atau *image* dan juga dapat menambah pengetahuan terapan dan mengaplikasikan ilmu steganografi yang telah didapat semasa kuliah.

b. Manfaat bagi Pengguna

   Dapat membantu penggunanya untuk mengirimkan informasi atau data-data yang bersifat rahasia agar sampai ke tangan penerima tanpa menimbulkan kecurigaan pada pihak lain. Pertama, informasi atau pesan rahasia terlebih dahulu diubah menjadi sandi khusus, sehingga pesan akan lebih terjaga kerahasiaannya. Kemudian, pesan yang telah terenkripsi disisipkan ke dalam media berupa gambar supaya pihak lain tidak menyadari bahwa terdapat pesan didalam media gambar tersebut.

c. Manfaat Universitas

   Dapat mengetahui kemampuan mahasiswanya dalam menguasai teori-teori dalam ilmu steganografi, keamanan data digital, dan dapat dijadikan bahan evaluasi.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Menurut Kamus Besar Bahasa Indonesia (KBBI), aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

Dari definisi diatas dapat disimpulkan bahwa aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas tertentu untuk pengguna.

Pengertian Aplikasi Menurut Para Ahli :

a. Aplikasi ialah sebuah aplikasi yang dimana tujuannya adalah agar bisa melayani setiap aktivitas komputerisasi yang dilakukan oleh pengguna. (Sari, 2017)
b. Pengertian software menurut (Syani, Werstantia, 2018) perangkat lunak adalah sebuah software yg berisi sebuah coding atau perintah yg dimana mampu diubah sesuai dengan harapan.

c. Menurut Wikipedia Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

2. *Steganography*

Steganografi atau *Steganography* adalah sebuah ilmu, teknik atau seni menyembunyikan sebuah pesan rahasia dengan suatu cara sehingga pesan tersebut hanya akan diketahui oleh si pengirim dan si penerima pesan rahasia tersebut. Steganografi berasal dari Bahasa Yunani yaitu Stegano yang berarti "tersembunyi atau menyembunyikan" dan graphy yang berarti "Tulisan, jadi Steganografi adalah tulisan atau pesan yang disembunyikan.Steganografi kebalikannya kriptografi yang menyamarkan arti dari sebuah pesan rahasia saja, tetapi tidak menyembunyikan bahwa ada sebuah pesan. Kelebihan Steganografi dibandingkan dengan Kriptografi adalah pesan-pesannya akan dibuat tidak menarik perhatian dan tidak menimbulkan kecurigaan, berbeda dengan Kriptografi yang pesannya tidak disembunyikan, walaupun pesannya sulit untuk dipecahkan akan tetapi itu akan menimbulkan kecurigaan pesan tersebut.

Terdapat tujuh teknik steganografi berdasarkan teknik yang digunakan, yaitu sebagai berikut (Ariyus,2009) :

1. *Injection*

   Merupakan teknik menanamkan pesan rahasia secara langsung terhadap suatu media, kekurangan dari teknik ini adalah media yang diinjeksi akan menjadi lebih besar dari ukuran normalnya sehingga mudah terdeteksi. Teknik ini juga disebut *embedding*.

2. *Substitution*

   Teknik ini mengubah data normal menjadi data rahasia, hasil dari teknik ini biasanya tidak akan terlalu mengubah ukuran data asli tetapi tergantung pada data yang akan disembunyikan. Teknik ini akan menurunkan kualitas media yang ditumpangi.

3. *Transformasi Domain*

   Teknik ini sangat efektif.Pada dasarnya, transformasi domain menyembunyikan data pada *transform space*.

4. *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 (*bandwidth*) 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*

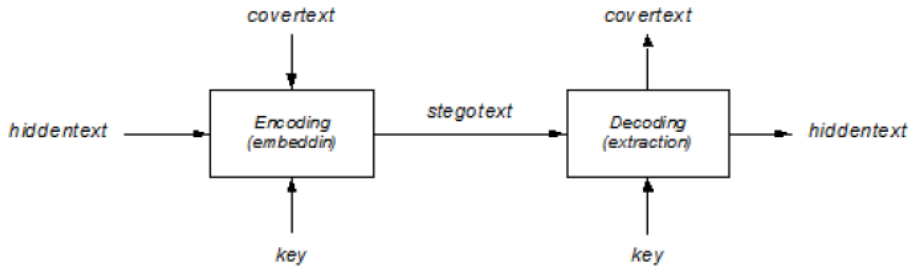
   Teknik ini menciptakan perubahan atas benda yang ditumpangi oleh data rahasia.

7. *Cover Generation*

   Teknik ini lebih baik daripada metode lainnya karena *cover* objek dipilih untuk menyembunyikan pesan.

Prinsip Kerja Steganografi

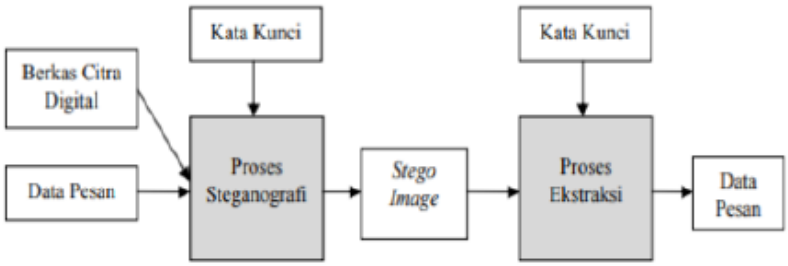
Pesan rahasia yang akan disembunyikan akan disisipkan pada suatu media penampung seperti citra, suara, video dan sebagainya yang terlihat tidak mencurigakan untuk menyimpan pesan rahasia. Pesan rahasia akan memerlukan sebuah kunci rahasia yang dinamakan *stego-key* agar hanya pihak yang berhak saja yang dapat membuka atau mengekstrak pesan rahasia tersebut (Immerse Lab, 14 -02-2018).



Gambar 2.1.*Diagram Encoding* dan *Decoding*
Sumber : (Jurnal ComTech, 2012)

Secara garis besar proses steganografi terdiri atas 2 bagian utama, yaitu proses penyembunyian data (*encode*) dan proses pengembalian data ke bentuk semula (*decode*). Kedua proses ini dilakukan dengan menggunakan sebuah kata kunci rahasia (*secret key*) yang akan digunakan di dalam prosesnya untuk

meningkatkan keamanan data. Hasil yang didapat setelah proses *embedding* atau *encoding* disebut *StegoObject.*



Gambar 2.2. Model Sistem *Steganografi*
Sumber : (Jurnal)

3. Enkripsi dan Dekripsi

Enkripsi (*encryption*) adalah sebuah proses menjadikan pesan yang dapat dibaca (*plaintext*) menjadi pesan acak yang tidak dapat dibaca (*ciphertext*). Berikut ini adalah contoh enkripsi yang digunakan oleh Julius Caesar, yaitu masing-masing huruf dengan 3 huruf selanjutnya disebut juga *Addition/Substitution Cipher.*

Plaintext	Ciphertext
rumah	dgymt
motor	yafad
kompor	waybad
dst...	...

Gambar 2.3. Contoh Enkripsi
Sumber : (Jurnal Prosiding SENDU_U, 2019)

Dekripsi merupakan proses kebalikan dari enkripsi dimana proses ini akan mengubah *ciphertext* menjadi *plaintext* dengan menggunakan algoritma pembalik dan *key* yang sama (Lekso Budi Handoko, Chaerul Umam, 2019).

Ciphertext	Plaintext
dgymt	rumah
yafad	motor
waybad	kompor
dst...	...

Gambar 2.4. Contoh Dekripsi
Sumber : (Jurnal Prosiding SENDU_U, 2019)

Enkripsi dan dekripsi adalah teknik pengkodean atau *decoding* untuk menghindari penyalahgunaan data jika itu terjadi pada orang yang salah.Ini melibatkan 2 kunci, publik dan pribadi. Setiap penerima atau pengirim mengetahui kunci privasinya tidak dapat diketahui oleh orang lain dan setiap kunci publik penerima atau pengirim diketahui oleh semua orang di jaringan. Dengan sebuah contoh, pengirim A ingin mengirim beberapa informasi ke penerima B. Ada orang lain 'C' yang mencari informasi di jaringan. Jadi, tujuan dari enkripsi adalah untuk menyandikan data sedemikian rupa sehingga hanya B yang dapat mendekode dan C tidak bisa. Kunci publik B diketahui semua orang di jaringan, jadi A menyandikan data menggunakan kunci publik B dan mengirimkannya ke jaringan. Kunci publik dan pribadi dari pengguna tertentu terkait secara matematis.Jadi, hanya jika anda menggunakan kunci pribadi orang tertentu anda dapat mendekode data yang telah dikodekan menggunakan kunci publik orang tersebut. Ketika B menerima data, ia menerjemahkannya menggunakan kunci privatnya sendiri, sehingga data yang dimaksudkan untuk B mencapai B. Misalkan C juga menerima data, tetapi ia tidak akan memiliki kunci *private* B sehingga ia tidak dapat mendekodenya (Dzul Fahmi Faisal Fadilah, Raufi Baihaqi Syahruzah, 2020).

4. *Citra* Digital

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/diskrit nilai digital yang disebut *pixel/picture elements*.*Citra* digital dapat dibedakan menjadi dua, yaitu *raster* dan *vector*.Pada umumnya, yang disebut dengan citra digital adalah citra digital dalam bentuk raster atau yang biasa disebut dengan citra bitmap (Al Caruban, 2018).



Gambar 2.5.*Citra* Digital
Sumber : (Pemrograman Matlab, 2017)

Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal *video* seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik.*Citra* digital merupakan suatu larik dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar.Jadi informasi yang terkandung bersifat diskrit.*Citra* digital tidak selalu merupakan hasil langsung data rekaman suatu sistem. Kadang-kadang hasil rekaman data bersifat kontinu seperti gambar pada monitor televisi, foto, sinar-*X*, dan lain sebagainya. Dengan demikian untuk mendapatkan suatu *citra* digital diperlukan

suatu proses konversi, sehingga citra tersebut selanjutnya dapat diproses dengan komputer (Achmad Ardiansyah, Mepa Kurniasih, 2018).

5. Metode *Spread Spectrum*

Metode *Spread Spectrum* dalam steganografi di ilhami dari skema komunikasi *spread spectrum,* yang mentransmisikan sebuah sinyal pita sempit ke dalam sebuah kanal pita lebar dengan penyebaran frekuensi. Fungsi dari penyebaran ini dimaksudkan untuk menambah redudansi (pengulangan) bit-bit data sehingga diharapkan dapat meningkatkan *robustness.*Adapun besaran redudansi ditentukan oleh faktor pengali yang sudah ditentukan oleh pengguna sendiri.Faktor pengali dilambangkan sebagai *variable* cari yang bernilai skalar. Panjang bit-bit hasil penyebaran akan menjadi cr kali panjang bit-bit awal. Proses pertama yang dilakukan terhadap pesan rahasia dalam metode *Spread Spectrum* adalah dengan mengubah pesan rahasia ke dalam bentuk string biner. Contoh :

made

m nilai ascii 109 string biner 0110 1101

a nilai ascii 97 string biner 0110 0001

d nilai ascii 100 string biner 0110 0100

e nilai ascii 101 string biner 0110 0101

digabungkan menjadi :

0110 1101 0110 0001 0110 0100 0110 0101

Hasil dari proses tersebut disisipkan ke dalam bit citra *red green blue* yang masing berjumlah 8 bit tergantung dengan hasil yang diperoleh dari data raster original pada sebuah gambar. Setelah melakukan proses penyisipan, maka

dalam tahap ini dilakukan proses ektraksi terhadap hasil modulasi. Setelah melakukan penyaringan, maka didapatkan bit-bit baru :

0110 1101 0110 0001 0110 0100 0110 0101

Spread Spectrum Steganography terpencar-pencar sebagai pesan yang diacak (*encrypt*) melalui gambar.Untuk membaca suatu pesan, penerima memerlukan algoritma yaitu *crypto-key* dan *stego-key*.Metode ini juga masih mudah diserang yaitu penghancuran atau pengrusakan dari kompresi dan proses *image* (gambar).

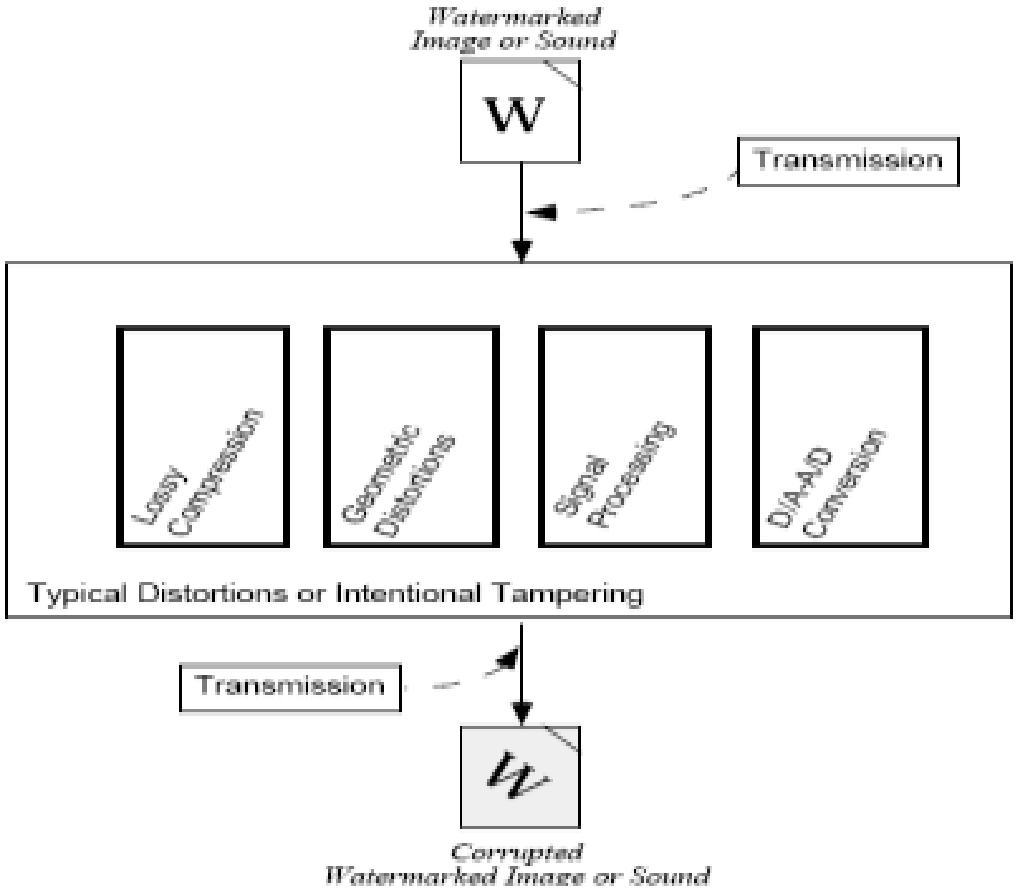
Pada proses penyembunyian data, bit-bit informasi yang telah mengalami proses *spreading* ini kemudian akan dimodulasi dengan *pseudo-noise signal* yang dibangkitkan secara acak berdasarkan kunci penyembunyian. Hasil dari proses modulasi ini kemudian digabungkan sebagai *noise* ke dalam sebuah berkas media pada bit-bit terakhir dari berkas media. Oleh penerima, sinyal dikumpulkan kembali menggunakan replika *pseudo-noise signal* untuk mendapatkan bit-bit yang berkorelasi.Bit-bit yang berkorelasi tersebut dianalisis dengan perhitungan tertentu untuk menghasilkan bit-bit informasi yang sesungguhnya.

Proses ekstraksi pesan menggunakan Metode *Spread Spectrum* ini terdiri dari tiga proses, yaitu pengambilan pesan dari matriks frekuensi, *demodulasi*, dan *dispreading*(Sri Hartati Monalisa, 2015).

Berdasarkan definisi, dapat dikatakan bahwa steganografi menggunakan metode *spread spectrum* memperlakukan *cover-object* sebagai derau (*noise*) ataupun sebagai usaha untuk menambah derau semu (*pseudo-noise*) ke dalam *cover-object*. *Cover-object* sebagai derau dapat menambahkan sebuah nilai ke

dalam *cover-object*.Nilai ini harus ditransmisikan di bawah tingkat derau yang ditambahkan nilai ke dalamnya.Hal ini berarti kapasitas sangat ditentukan oleh *cover-object*(Widianto, 2017).

Adapun cara kerja dari algoritma *spread spectrum* :



Gambar 2.6. Algoritma *Spread Spectrum*
Sumber : (Jurnal Telematik, 2015)

6. PHP (*Hypertext Preprocessor*)

Menurut Supono & Putratama (2018:1) mengemukakan bahwa "**PHP** (**PHP**: *hypertext preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang

dapat dimengerti oleh komputer yang bersifat *server-side* yang ditambahkan ke HTML".

PHP menurut Mundzir (2018:3) PHP yaitu bahasa pemrograman *universal*untuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan bersamaan dengan HTML.

PHP merupakan *software Open-Source* yang disebarkan dan dilisensikan secara gratis serta dapat di *download*secara Bahasa dari situs resminya http://www.php.net PHP ditulis menggunakan Bahasa C. PHP memiliki banyak kelebihan yang tidak dimiliki oleh Bahasa *script* sejenis. PHP difokuskan pada pembuatan *script server-side,* yang biasa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data *form,* menghasilkan isi halaman web dinamis, dan kemampuan mengirim serta menerima *cookies,* bahkan lebih daripada kemampuan CGI.

7. *XAMPP*

Menurut Aryanto (2016:4), "*XAMPP* merupakan sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman seperti :*Apache, HTTP, MySQL, database*, bahasa pemrograman *PHP* dan *Perl*."

Menurut Iqbal (2019:15), "*XAMPP* merupakan sebuah *software web server apache* yang didalamnya sudah tersedia *database server MySQL* dan *support PHP programming*".

Berdasarkan pengertian diatas dapat disimpulkan bahwa *XAMPP* adalah sebuah aplikasi perangkat lunak yang di dalamnya sudah tersedia *database server MySQL* dan *support PHP programming*.

8. *Flowchart*

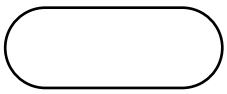
Flowchart merupakan gambaran atau bagan yang memperlihatkan urutan dan hubungan antara proses beserta instruksinya. Gambar ini dinyatakan dengan simbol. Dengan demikian setiap *symbol* menggambarkan proses tertentu. Sedangkan hubungan antara proses digambarkan dengan garis penghubung.

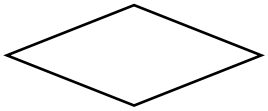
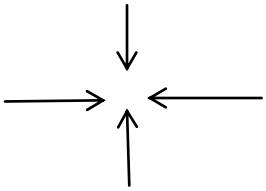
Flowchart ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan proses kegiatan menjadi lebih jelas, Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya program (*programmer*) menerjemahkannya ke bentuk program dengan bahasa program.

Flowchart disusun dengan simbol-simbol. Simbol ini di pakai dengan alat bantu menggambarkan proses didalam program.

Simbol-simbol yang di pakai antara lain :

Tabel 2.1.*Flowchart*

No	Simbol	Fungsi
1		Terminal untuk memulai dan mengakhiri suatu program.
2		Proses suatu *Symbol* yang menunjukkan setiap pengolahan yang di lakukan oleh komputer.

No	Simbol	Fungsi
3		*Input / output*, untuk memasukkan data maupun menunjukkan hasil suatu proses.
4		*Decision* suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan.
5		Arus bawa *follow*, prosedur yang dapat dilakukan atas bawa, bawa ke atas, dari kanan, atau dari kanan kekiri.

9. *UML*

Unified Modelling Language (*UML*) adalah sebuah "bahasa" yang telah menjadistandar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML*menawarkan sebuah standar untuk merancang model sebuah sistem.

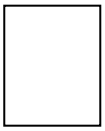
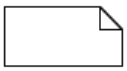
Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.Tetapi karena *UML* juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti *C++*, *Java*, *C#* atau *VB.NET*. Walaupun demikian, *UML*tetap dapat digunakan untuk modeling aplikasi prosedural dalam *VB* atau *C*.

Seperti bahasa-bahasa lainnya, *UML*mendefinisikan notasi dan *syntax*/semantik. Notasi *UML*merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UMLsyntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari 3 notasi yang telah ada sebelumnya : Grady Booch *OOD* (*Object-Oriented Design*), Jim Rumbaugh *OMT*(*Object Modeling Technique*), dan Ivar Jacobson *OOSE* (*Object-Oriented Software Engineering*).

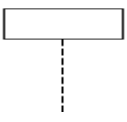
Adapun daftar simbol *UML*yaitu :

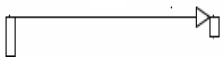
Tabel 2.2. Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)*akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.

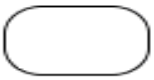
NO	GAMBAR	NAMA	KETERANGAN
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*Sistem*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan 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 perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2.3. Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.

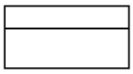
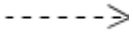
NO	GAMBAR	NAMA	KETERANGAN
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

Tabel 2. 4. Simbol *Activity Diagram*

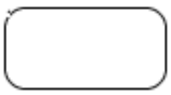
NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

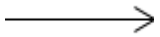
Tabel 2. 5. Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).

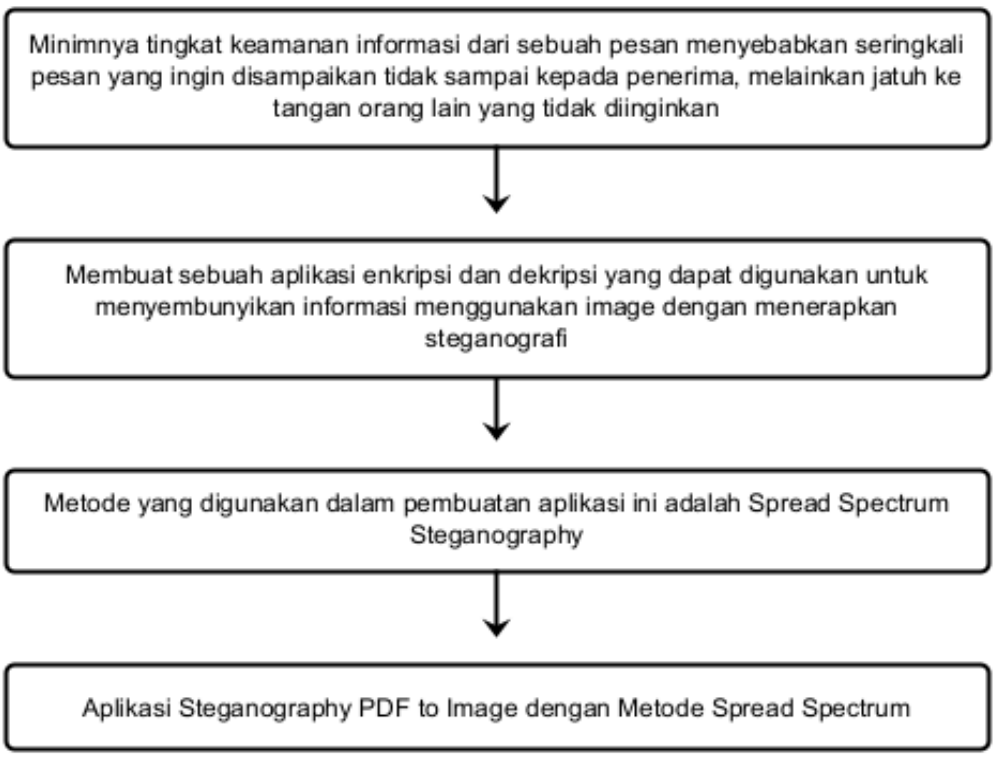
NO	GAMBAR	NAMA	KETERANGAN
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang 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
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 6.*State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan

NO	GAMBAR	NAMA	KETERANGAN
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

10. Kerangka Fikir

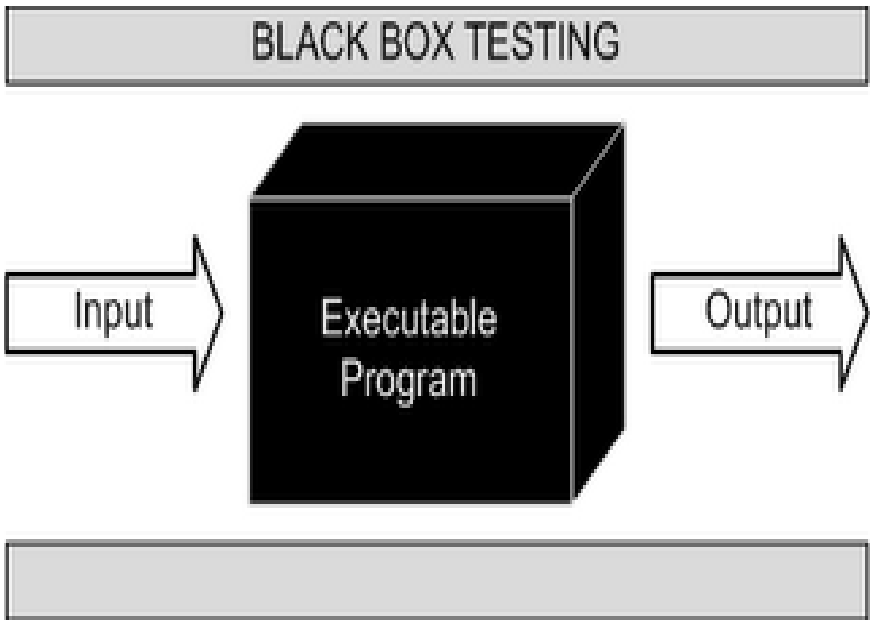


11. Pengujian *Black Box*

Black BoxTesting adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat. Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program(Rosa, Salahuddin, 2015)

Pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut :

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



Gambar 2. 7.*Blackbox* Testing
Sumber : (http://tkjpnup.blogspot.com/,2013)

Contoh pengujian *Black Box* :

Berdasarkan kasus pengujian pada salah satu aplikasi dalam hal ini misalnya nama admin yang benar yaitu “admin” dan kata sandinya yaitu “123”

Tabel 2. 7. Contoh Pengujian *Blackbox*

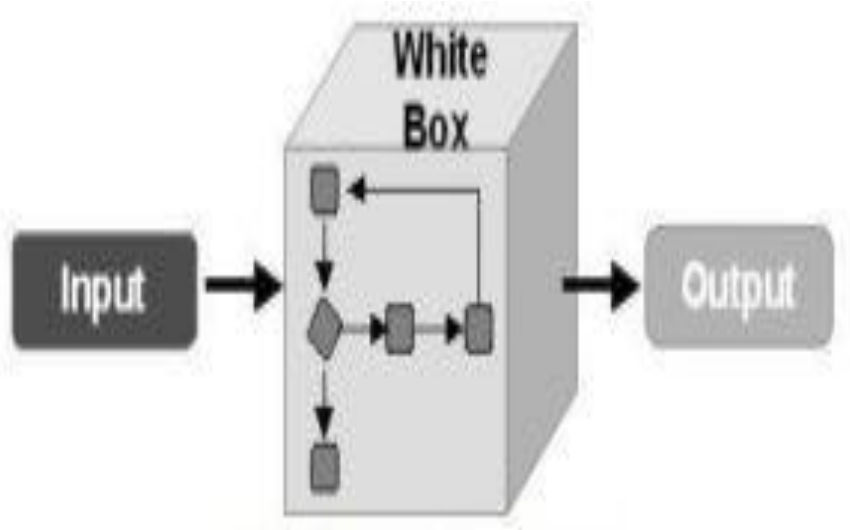
No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Status
1.	Mengosongkan semua isian data *Login*, lalu langsung mengklik tombol ‘Masuk’.	Nama: - Kata sandi: -	Sistem akan menolak akses *Login* dan menampilkan pesan “Mohon isi dulu nama admin dan kata sandi”	Sesuai harapan	Valid
2.	Hanya mengisi data nama admin dan mengosongkan data kata sandi, lalu mengklik tombol ‘Masuk’.	Nama: admin Kata sandi: -	Sistem akan menolak akses *Login* dan menampilkan pesan “Mohon isi dulu salah satu data yang masih kosong”	Sesuai harapan	Valid
3.	Hanya mengisi data kata sandi dan mengosongkan data nama admin, lalu langsung mengklik tombol ‘Masuk’.	Nama: - Kata sandi: 123	Sistem akan menolak akses *Login* dan menampilkan pesan “Mohon isi dulu salah satu data yang masih kosong”	Sesuai harapan	Valid

No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Status
4.	Menginputkan dengan kondisi salah satu data benar dan satu lagi salah, lalu langsung mengklik tombol ‘Masuk’.	Nama: mimin (salah). Kata sandi: 123 (benar).	Sistem akan menolak akses *Login* dan menampilkan pesan “Nama pengguna atau kata sandi salah! Silakan ulangi dengan data yang benar”	Sesuai harapan	Valid
5.	Menginputkan sintaks SQL Injection pada kedua data *Login*, lalu mengklik tombol ‘Masuk’.	Nama: ‘ OR ’1′=’1 Kata sandi: ‘ OR ’1′=’1	Sistem akan menolak akses *Login* dan menampilkan pesan “Nama pengguna atau kata sandi salah! Silakan ulangi dengan data yang benar”	Sesuai harapan	Valid
6.	Menginputkan data *Login* yang benar, lalu mengklik tombol ‘Masuk’.	Nama: admin Kata sandi: 123	Sistem menerima akses *Login* dan kemudian langsung menampilkan form pakar/admin.	Sesuai harapan	Valid

12. Pengujian *White Box*

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau perangkat lunak dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulangdan diperiksa lagi sampai Anda mencapai apa yang

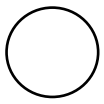
diharapkan pengujian yang didasarkan pada kontrol detail desain. Struktur kontrol desain program digunakan dengan cara prosedur untuk membagi tes ke dalam berbagai kasus uji. Sekilas dapat disimpulkan bahwa teks kotak putih adalah panduan untuk membuat program 100% benar.(Akbar Asfihan, 2019)



Gambar 2. 8.*Whitebox* Testing
Sumber : (raharja.ac.id,2020)

A. Istilah-istilah dalam pembuatan *flowgraph*

a. Node yaitu lingkaran pada *flowgraph* yang menggambarkan satu atau lebih perintah prosedural. Urutan proses dan keputusan dapat dipetakan dalam satu node.



b. Edge yaitu tanda panah yang menggambarkan aliran kontrol. Setiap node harus mempunyai tujuan node.

c. *Region* yaitu daerah yang dibatasi oleh node dan edge. Untuk menghitung *region* daerah diluar *flowgraph* juga harus dihitung.



d. Predicate Node yaitu kondisi yang terdapat pada node dan mempunyaikarakteristik dua atau lebih edge darinya.



e. *CyclomaticComplexity* yaitu metric perangkat lunak yang menyajikanukuran kuantitatif dari kekompleksan logika suatu program. *CyclomaticComplexity* digunakan untuk mencari jumlah data dalam suatu*flowgraph*.

f. *Independent Path* yaitu jalur yang melintasi atau melalui dalam program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru. Dalam istilah *flowgraph, independentpath* harus bergerak sekurang-kurangnya pada satu edge, dimana edge yang akan dilewati tidak dapat dilewati edge sebelum jalur tersebut didefinisikan.

B. Teknik Pengujian

Untuk menguji suatu sistem maka bagan alir program (*Flowchart*) yang didesain sebelumnya dipetakan ke dalam bentuk bagan alir control (*flowgraph*) yang nantinya memudahkan untuk penentuan jumlah region,*Cyclomatic Complexity (CC)* dan *Independent Path*, jika jumlah region, *Cyclomatic Complexity (CC)* dan *Independent Path* sama besar makasistem dinyatakan benar, tetapi jika sebaliknya maka sistem masih memiliki kesalahan, mungkin dari segi logika maupun dari sisi lainnya. *Cyclomatic Complexity* (CC) dapat dihitung dengan menggunakan rumus:

$$V(G) = E - N + 2$$

Dimana : E = Jumlah *edge* pada *flowgraph*

N = Jumlah *node* pada *flowgraph*

Dengan menggunakan metode *White Box testing* penulis dapat melakukan *test case* yaitu:

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
2. Mengerjakan semua keputusan logika pada sisi *true* dan f*alse*.
3. Mengeksekusi semua *loop* pada batasannya dan pada operasionalnya.
4. Menggunakan struktur dana internal untuk menjamin validitasnya.

C. Manfaat

Adapun manfaat yang dapat diperoleh dengan mengunakan teknik pengujian *White Box* adalah antara lain sebagai berikut:

1. Dapat menjamin seluruh independent path di dalam modul yang dikerjakan sekurang-kurangnya satu kali.
2. Dapat mengerjakan seluruh keputusan logikal.
3. Dapat mengerjakan seluruh perulangan yang sesuai dengan batasannya.
4. Dapat mengerjakan seluruh struktur data internal yang menjamin validilitas.

Login

Node 1 : Mulai

Node 2 : Masukkan status *admin, username, password*

Node 3 : Kondisi “IF” jika hasil pilihan = “Y” ke Node 4, jika = “T” ke Node 5

Node 4 : Menu Utama

Node 5 : Kondisi “IF” jika hasil pilihan = “Y” ke Node 5, jika = “T” ke Node 2

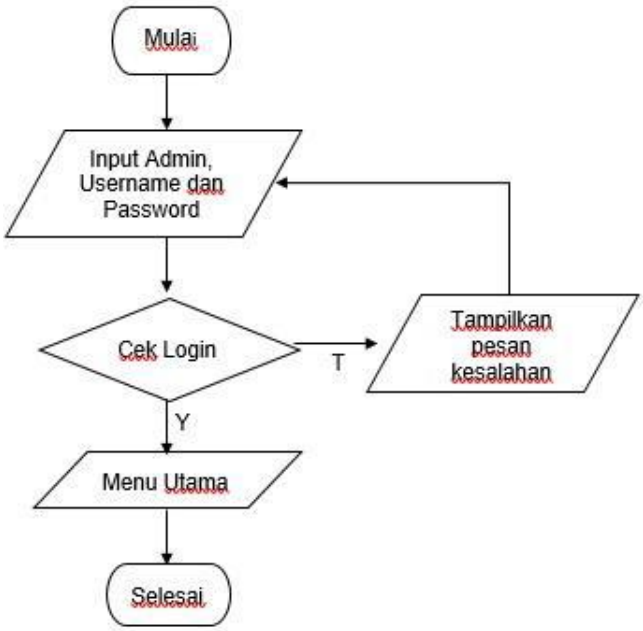
Node 6 : Selesai

Independent Path = 3, yaitu

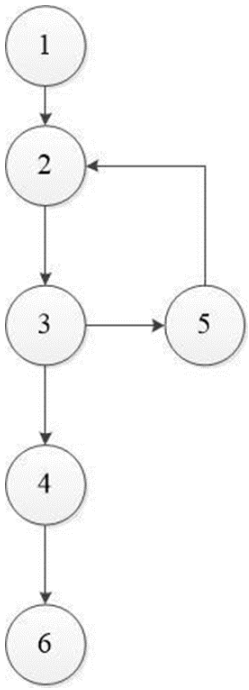
Path 1 = 1-2-3-4-6

Path 2 = 1-2-3-4-5-6

Path 3 = 1-2-3-5-2



Gambar 2. 9.*Flowchart Whitebox*
Sumber : (skripsi,2020)



Gambar 2. 10.*Flowgraph Whitebox*
Sumber : (raharja.ac.id,2020)

B. Kajian Hasil Penelitian Terdahulu

1. Penelitian ini ditulis oleh Nurwahida Milanda pada tahun 2020 dengan judul **"Aplikasi Perpaduan Algoritma Enkripsi *RC4* dan *BASE64* dengan Metode Steganografi *LSB* menggunakan Bahasa Pemrograman *PHP*".** Pada penelitian ini penulis membuat aplikasi penggabungan antara teknik kriptografi dan teknik steganografi menggunakan sebuah penyembunyian informasi. Metode algoritma yang digunakan adalah *RC4* dan *Base64* dalam proses enkripsi, dikombinasikan dengan menyisipkannya pada gambar menggunakan metode *LSB*. Berdasarkan pengujian yang dilakukan, dihasilkan sebuah aplikasi yang dapat menyembunyikan informasi dengan menggunakan gambar sebagai penyembunyi yang berformat (**jpg*). Hasil dari penelitian ini adalah berupa perubahan ukuran gambar, yaitu ukuran gambar asli dibandingkan dengan ukuran gambar sesudah disisipkan pesan yang terenkripsi berdasarkan penyisipan karakter kelipatan 100 (dari 100 karakter hingga 1000 karakter).
2. Penelitian ini ditulis oleh Asri pada tahun 2020 dengan judul **"Implementasi Algoritma Kriptografi Elgamal untuk Proses Enkripsi dan Dekripsi Guna Pengamanan *File* Data".** Pada penelitian ini penulis mencoba mengimplementasikan algoritma *ElGamal* untuk mengenkripsi sebuah *file* bukan hanya sekedar *file* berekstensi *txt*, namun penulis mencoba mengimplementasikan pada *file* berekstensi *docx*. Dalam penelitian ini penulis mengambil subjek di Media Center UM Parepare agar kerahasiaan

dan keamanan dokumen-dokumen lembaga tersebut terutama yang berekstensi *docx* dapat terjaga.

3. Penelitian ini ditulis oleh Andi Haerul pada tahun 2019 dengan judul **"Aplikasi Steganografi Menggunakan Metode *LSB* (*Least Significant Bit*) pada Media *Audio*".** Penulis membuat aplikasi steganografi yaitu dengan menyembunyikan *file* teks pada *cover audio Wav*, penelitian ini menghasilkan aplikasi yang dapat menyembunyikan informasi *file* teks dengan *cover audio Wav* dengan ukuran *fileaudio,bitrate*, dan panjang audionya tetap sama sesuai dengan *file audio* yang aslinya, *file audio* dengan menggunakan metode *Least Significant Bit* (LSB). Informasi yang disembunyikan dalam *cover audio* kualitas informasinya tidak mengalami pengurangan dan dapat menjamin kerahasiaan sebuah informasi yang disembunyikan dalam *coveraudio* karena dimensi audio sebelum dan sesudah disisipkan *file* teks tidak mengalami perubahan.

Berbeda dengan 3 penelitian diatas, penulis mengangkat judul "Aplikasi *Steganography*PDF to Image dengan metode *Spread Spectrum*". Penelitian ini membahas tentang enkripsi *filepdf* ke gambar sehingga orang ketiga tidak dapat membuka pesan tersebut. Bahasa pemrograman yang digunakan adalah *PHP*.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang dilakukan merupakan penelitian eksperimental, yaitu penelitian yang pengumpulan datanya melalui pencatatan secara langsung dari hasil percobaan yang dilakukan.

B. Lokasi dan Waktu Penelitian

Tempat penelitian dilakukan di Universitas Muhammadiyah Parepare. Waktu yang dibutuhkan dalam pelaksanaan penelitian ini adalah ± 4 bulan.

C. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut. Berikut ini merupakan penjelasan dari *hardware* dan *software* yang digunakan dalam pembuatan aplikasi *Steganography* ini.

1. Perangkat Keras (*Hardware*)
    a. *Laptop* : *Acer Aspire E14*
    b. *Processor* : *Intel® Core™ i5-4210U @ 1.70GHz 2.40 GHz*
    c. *RAM* : *4 GB*
    d. *Harddisk* : *500 GB*
    e. *OS* : *Windows 10 64 bit*

2. Perangkat Lunak (*Software*)
    a. *Xampp*
    b. *SublimeText 3*
    c. *Browser Google Chrome*

D. Teknik Pengumpulan Data

1. Secara tidak langsung (*Studi Literatur*)

   Metode tidak langsung ini maksud ialah mengumpulkan data-data tentang *Steganography* maupun informasi yang terkait seperti artikel yang berasal dari media internet.

2. Secara langsung (*Observasi*)

   Metode secara langsung yaitu mengumpulkan data-data atau informasi yang terkait dengan perancangan program aplikasi *Steganography*.

E. Tahap Penelitian

Tahap-tahap penelitian yang dimaksud dalam penelitian ini adalah berkenan dengan proses pelaksanaan penelitian.

1. Persiapan penelitian

   Tahap persiapan adalah tahap yang dilakukan sebelum melakukan penelitian. Pada tahap ini dimulai dengan mengkaji permasalahan aplikasi *steganography* enkripsi pdf *to image*.

2. Tahapan Implementasi Sistem

   Tahap implementasi sistem menggambarkan tentang proses yang terjadi pada sistem. Rancangan penelitian diimplementasikan dalam perangkat lunak yang siap digunakan untuk *user*.

3. Pengujian Sistem

   Pengujian sistem dilakukan untuk melihat kinerja dari masing-masing komponen yang dipakai untuk membangun sistem guna mengetahui tingkat keberhasilan pada sistem yang dibuat.

4. Analisa Hasil

   Dari pengujian sistem yang dilakukan analisa kerja sistem dan data yang didapatkan selama pengujian.

5. Dokumentasi Hasil

   Dokumentasi hasil merupakan tahap akhir dari penelitian ini guna sebagai pelaporan hasil penelitian yang meliputi hasil pengujian sistem, program, foto-foto selama pengujian dan hal-hal lainnya.

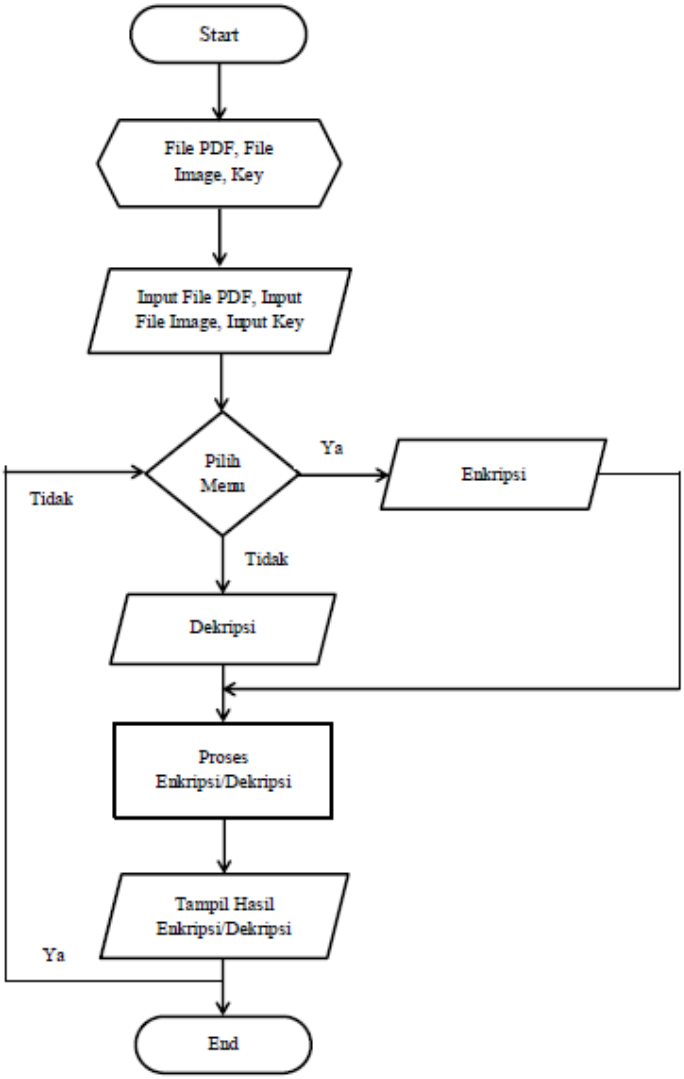
F. Metode Pengujian

Metode pengujian ini menggunakan :

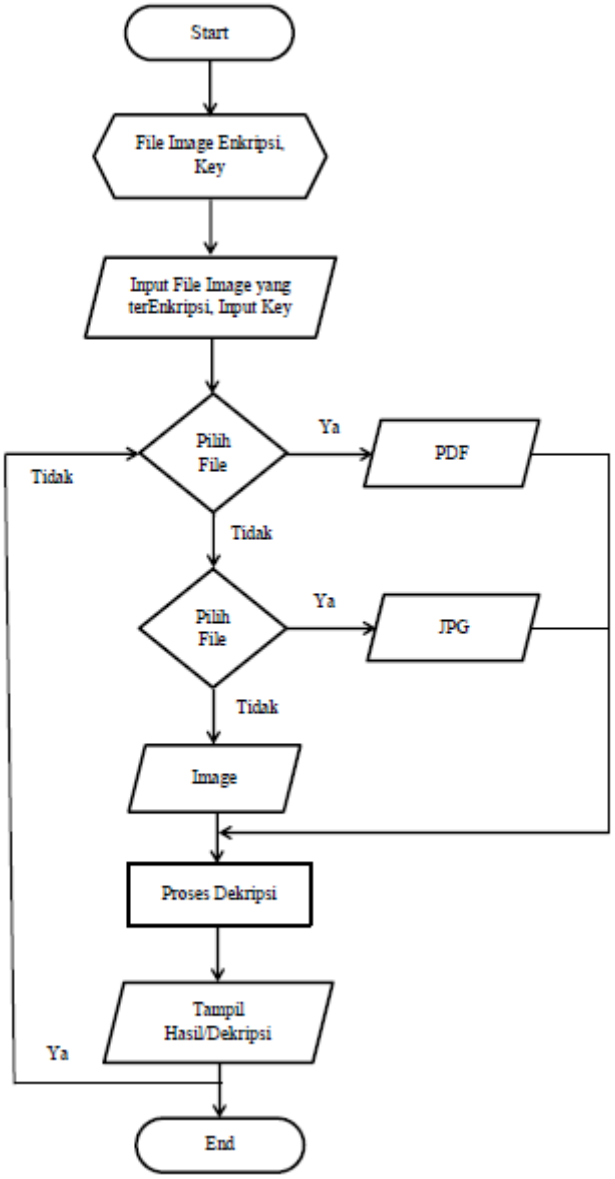
1. *Black box*, yaitu pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dengan kata lain pengujian *black box* mengevaluasi hanya dari tampilan luarnya (*interfacenya*), fungsionalnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui *input* dan *output*).
2. *White box*, yaitu pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

G. Desain Aplikasi

1. *Flowchart*Aplikasi

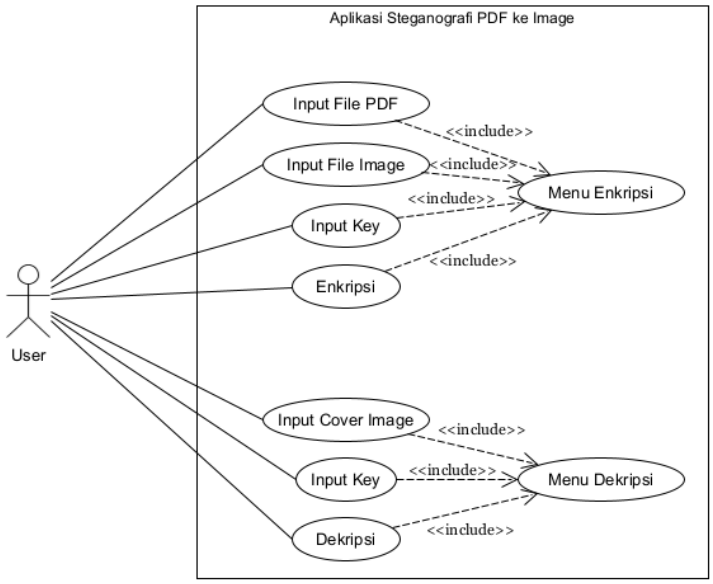


Gambar 3. 1*Flowchart Menu* Enkripsi



Gambar 3. 2.*Flowchart Menu* Dekripsi

2. *Use Case Diagram*



Gambar 3. 3.*Use Case Diagram*

Penjelasan *usecase diagram* :

Tabel 3. 1. Penjelasan *usecase* penelitian

Nama *Usecase*	Deskripsi *Usecase*
InputFilePDF	*Usecase* ini menjelaskan *user* menginputkan *file* berEkstensi *PDF* terlebih dahulu untuk *file* yang ingin di sisipkan atau di enkripsi
InputFileImage	*Usecase* ini menjelaskan *user* menginputkan *fileimage* berEkstensi *JPG,* untuk sebagai *cover* dari *file* yang disisipkan atau enkripsi
InputKey	*Usecase* ini menjelaskan *user* menginputkan *key* atau *password*, untuk mengamankan *file* yang telah terEnkripsi
Enkripsi	*Usecase* ini menjelaskan tentang proses Enkripsi *filePDF* ke *file Image*
InputCover Image	*Usecase* ini menjelaskan *user* menginputkan *coverimage* yang sudah terEnkripsi, untuk di dekripsi
InputKey	*Usecase* ini menjelaskan *user* menginputkan *Key* atau *Password*, untuk membuka keamanan dari *coverimage* yang telah terEnkripsi
Dekripsi	*Usecase* ini menjelaskan tentang proses Dekripsi *file cover image*

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Perancangan Sistem

Pada dasarnya aplikasi yang dibuat dalam tugas akhir ini adalah aplikasi Enkripsi dan Dekripsi *filePDF* ke *Image* menggunakkan metode *Sread Spectrum*yang terdiri dari 2 (dua) unsur utama yakni perangkat keras dan perangkat lunak, sehingga rancangan sistemnya terdiri dari rancangan keras dan rancangan perangkat lunak.

1. Rancangan perangkat keras

Adapun perangkat keras yang digunakan untuk mengimplementasikan rancangan pada penelitian ini adalah berupa laptop *Acer* dengan *spesifikasi Intel® Core™ i5-4210U @ 1.70GHz 2.40 GHz processor, RAM 4GB, Hard Disk 500 MB.*

2. Rancangan perangkat lunak

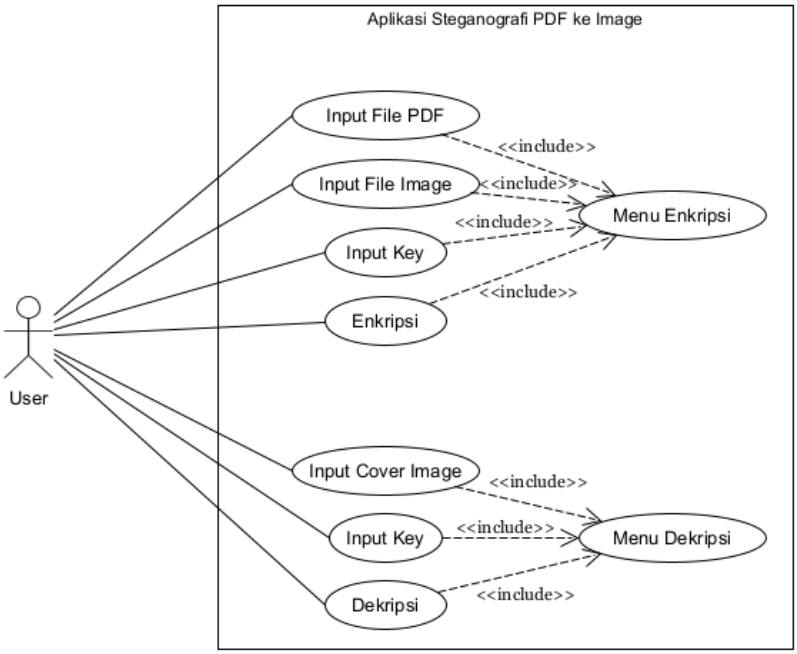
Adapun perangkat lunak yang digunakan untuk mengimplementasikan rancangan pada penelitian ini adalah untuk rancangan bangun perangkat lunak yaitu menggunakan sistem operasi *Windows 10 64 bit*, bahasa pemrograman *PHP*, dan *tools* seperti *xampp, SublimeText 3, browser* serta jenis *file* yang digunakan saat penelitian adalah *PDF dan Image*. Kemudian untuk perancangan sistem, perlu dilakukan agar memberikan gambaran yang jelas dan lengkap tentang rancangan bangun dan implementasi bagaimana sistem dibuat terdiri dari :

a. Perancangan model *UML* (*Unified Modelling Language*)

Sistem yang penulis rancang adalah sistem berupa enkripsi dan dekripsi *filepdf* dan *image.* Adapun dalam melakukan perancangan sistem adalah dengan memanfaatkan *Diagram UML* berupa *Usecase diagram, Activity diagram,* dan *Sequence diagram.*

1) Use case Diagram

Use Case diagram berfungsi untuk menjelaskan alur sistem jika dilihat menurut pandangan orang yang berada diluar sistem.

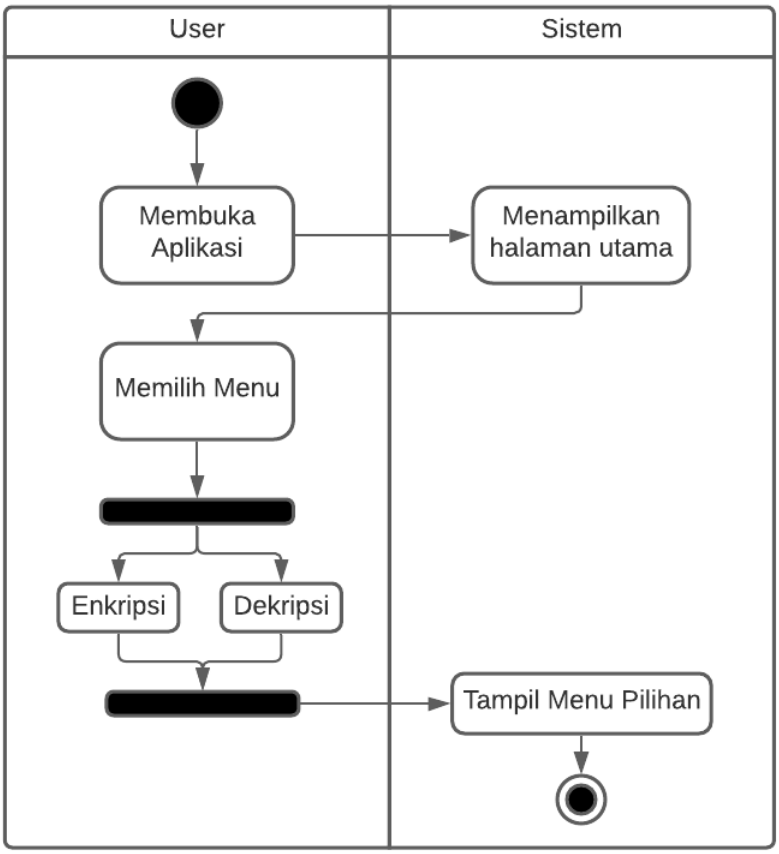


Gambar 4. 1. Rancangan *Usecase Diagram*

2) *Activity Diagram*

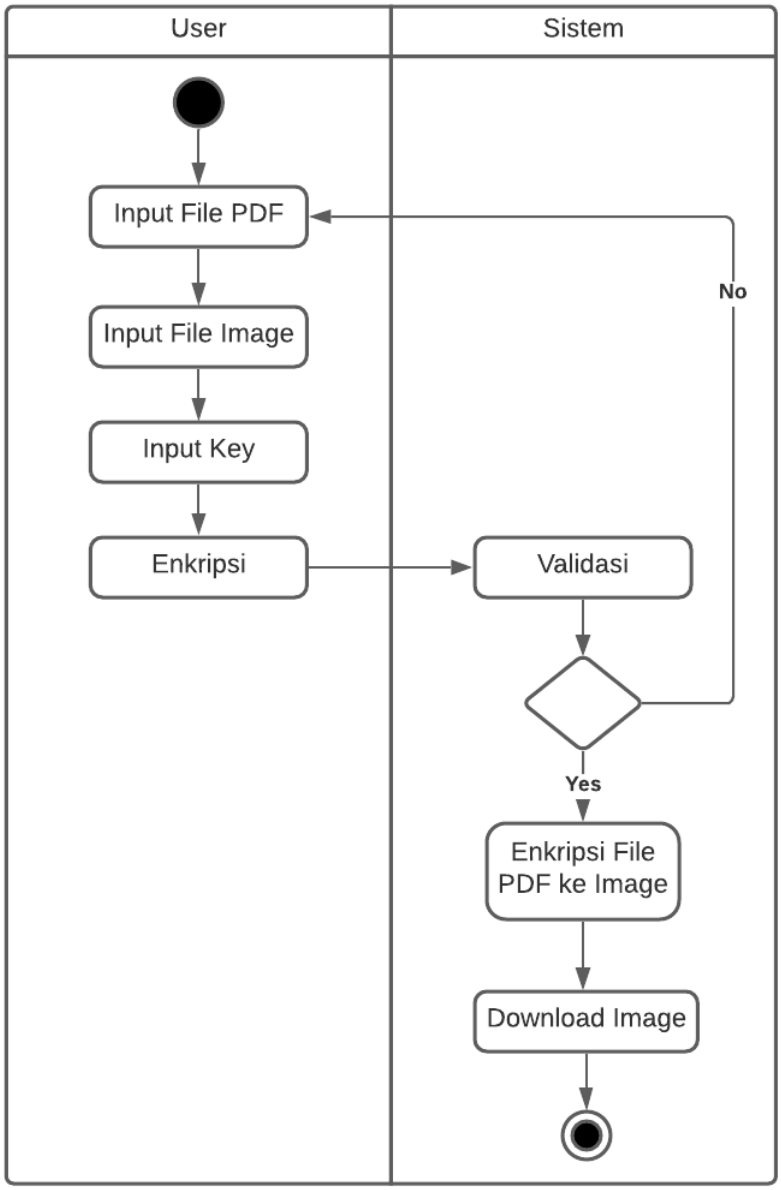
Activity diagram memodelkan alur kerja (*workflow*) sebuah proses dan urutan dalam suatu proses. *Activity diagram* pada sistem yang penulis rancang antara lain :

a) Rancangan *Menu*



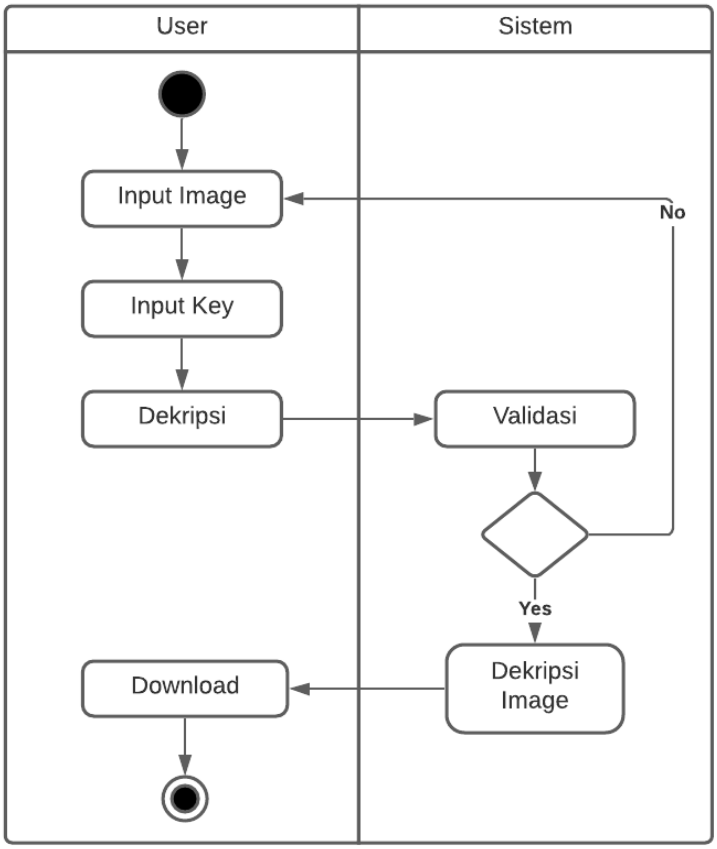
Gambar 4. 2.*Activity Diagram*Rancangan *Menu* Utama

b) Rancangan Enkripsi



Gambar 4. 3.*Activity Diagram*Rancangan *Menu* Enkripsi

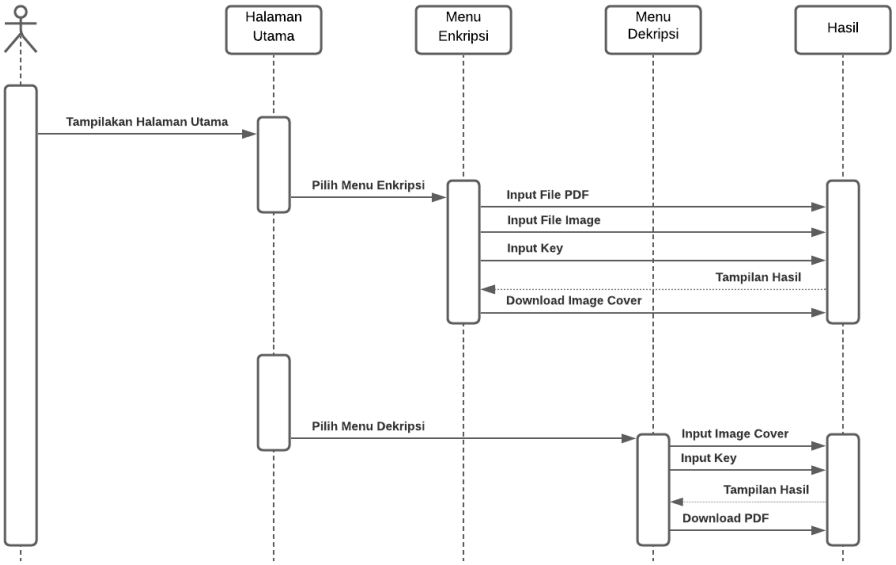
c) Rancangan Dekripsi



Gambar 4. 4.*Activity Diagram*Rancangan *Menu* Dekripsi

3) Sequence Diagram

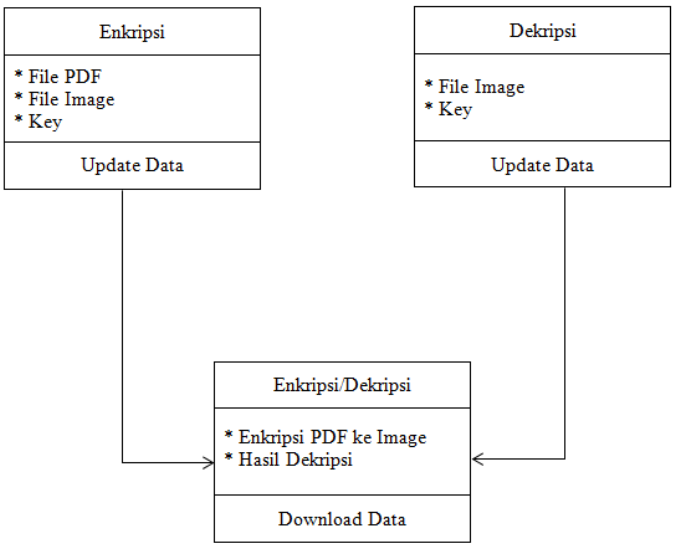
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antara objek dan mengkondisikan komunikasi antara objek-objek tersebut. Berikut *sequence diagram* aplikasi yang akan dibangun.



Gambar 4. 5.*Sequence diagram* aplikasi yang dibangun

4) Class Diagram

Class diagram adalah jenis diagram struktur statis yang menggambarkan struktur sistem dengan menunjukkan system *class*, atribut, metode, dan hubungan antar objek. Berikut *Class Diagram* yang dirancang :

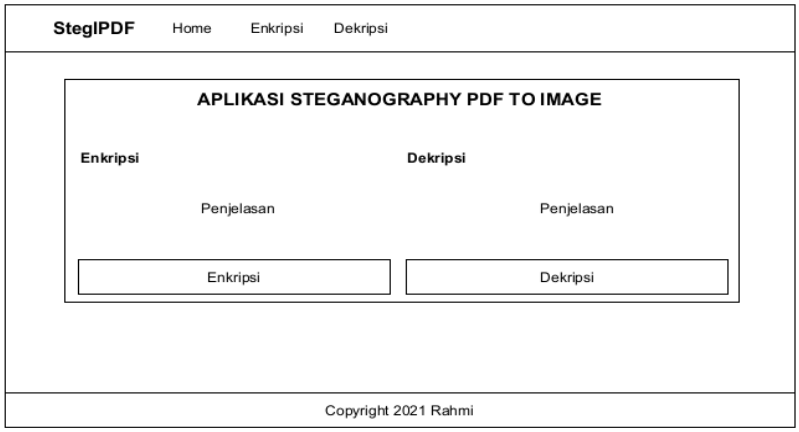


Gambar 4. 6.*ClassDiagram*

b. Perancangan *Interface*

Rancangan *interface* merupakan desain tampilan dari sebuah sistem yang bertujuan untuk menggambarkan sistem yang akan dibangun. Adapun rancangan *interface* untuk Aplikasi Steganografi pada file *pdf* ke *image* menggunakan metode *spread spectrum*, yaitu :

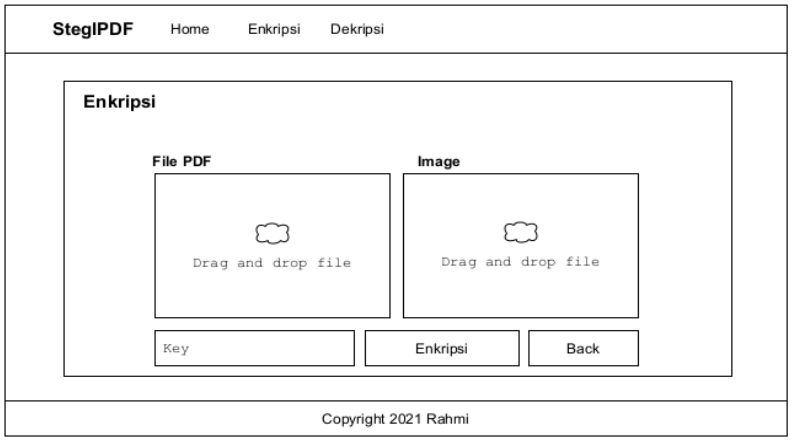
1) Rancangan *Interface* Halaman Utama



Gambar 4. 7. Rancangan *Interface* Halaman Utama

Halaman utama merupakan tampilan awal pada saat aplikasi ini di akses pertama kali, dengan menampilkan nama aplikasi dan definisi dari Enkripsidan Dekripsi.

2) Rancangan *InterfaceMenu* Enkripsi

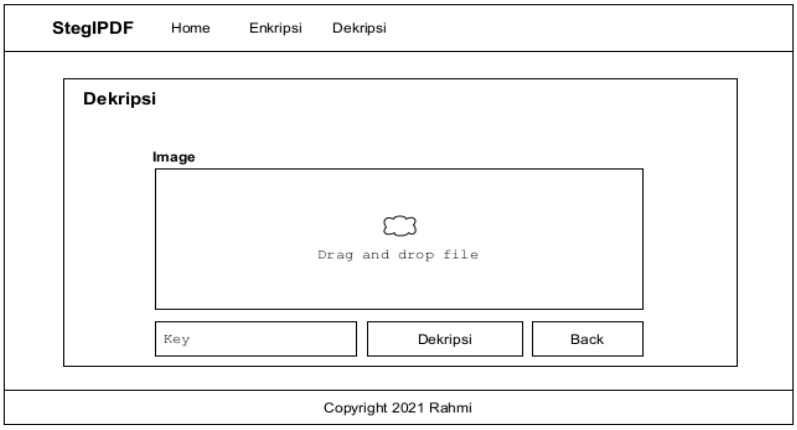


Gambar 4. 8. Rancangan *Interface Menu* Enkripsi

*Menu*Enkripsi merupakan tampilan yang berfungsi untuk melakukan proses enkripsi. Di dalam ini, terdapat fitur yaitu :

a) *Input PDF* : untuk mencari letak direktori *filepdf* yang akan di enkripsi.
b) *Input Image* : untuk mencari letak direktori *file image* yang akan dijadikan *cover image* enkripsi.
c) *Input Key* : untuk sebagai password keamanan enkripsi.
d) Enkripsi : untuk melakukan proses enkripsi pada *pdf* dan *image* yang sudah diinput
e) *Back* : untuk kembali ke *menu* utama

3) Rancangan *Interface Menu* Dekripsi



Gambar 4. 9. Rancangan *Interface Menu* Dekripsi

Menu Dekripsi merupakan tampilan yang berfungsi untuk melakukan proses dekripsi. Di dalam ini, terdapat fitur yaitu :

a) *Input Cover Image* : untuk mencari letak direktori *file image* yang sudah terEnkripsi.

b) *Input Key* : untuk membuka *password* keamanan dari *image cover* yang telah terEnkripsi.

c) Dekripsi : untuk melakukan proses dekripsi terhadap *file image cover* yang sudah diinput.

d) *Back* : untuk kembali ke menu utama.

B. Pembuatan Aplikasi

Adapun langkah-langkah dalam membuat Aplikasi*Steganography* Pada File *PDF to Image* dengan metode *Spread Spectrum*, yaitu :

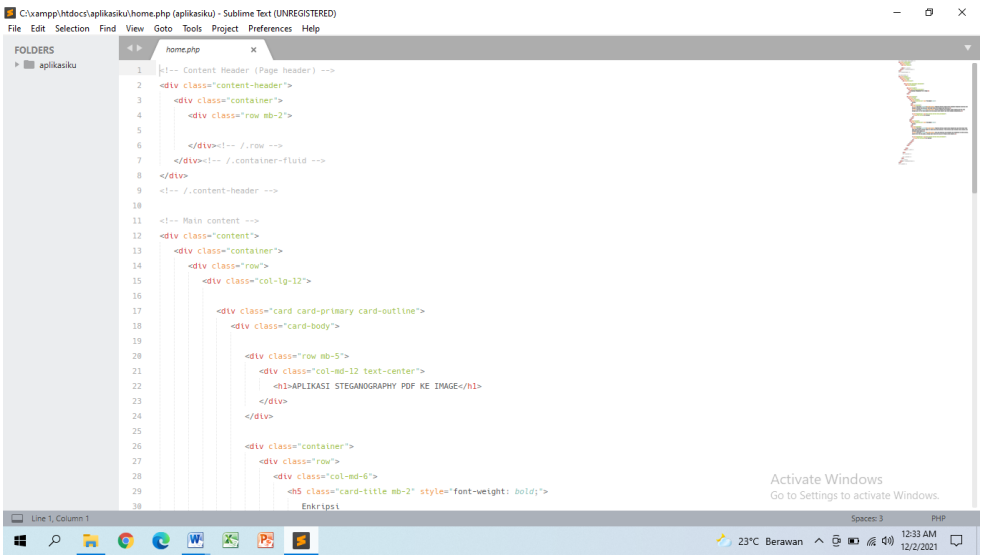
1. Semua *tools* yang digunakan telah ter*install* seperti text editor *Sublime Text*, *XAMPP* dan *Browser Google Chrome*.
2. Menjalankan *XAMPP* dan membuka text editor *Sublime Text*.
3. Pertama membuat Halaman Utama

   Halaman utama atau yang lebih dikenal sebagai *user interface* adalah media yang menghubungkan manusia dengan komputer agar dapat saling berinteraksi.



Gambar 4. 10. Halaman Utama

Untuk membuat Halaman utama seperti diatas, dapat mengetikkan kode seperti dibawah ini.



Gambar 4. 11.*Source Code* Menu Utama

C. Detail Aplikasi

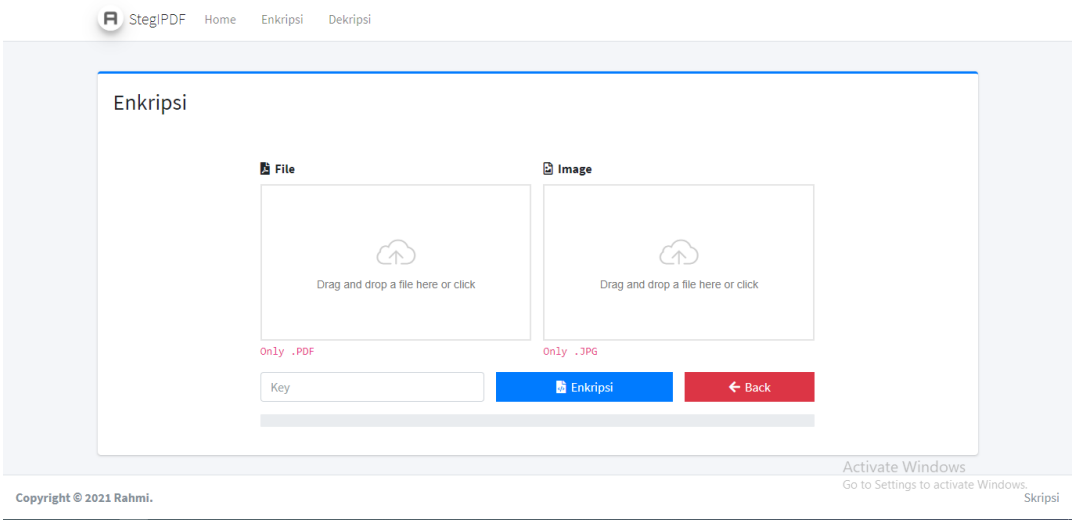
1. Halaman Utama



Gambar 4. 12. Halaman Utama Aplikasi

Halaman atau tampilan awal Aplikasi *Steganography pdf to image* dikunjungi. Adapun cara membuka halaman utama dengan menggunakan pendukung *XAMPP* yaitu : *localhost/stegano-8*.

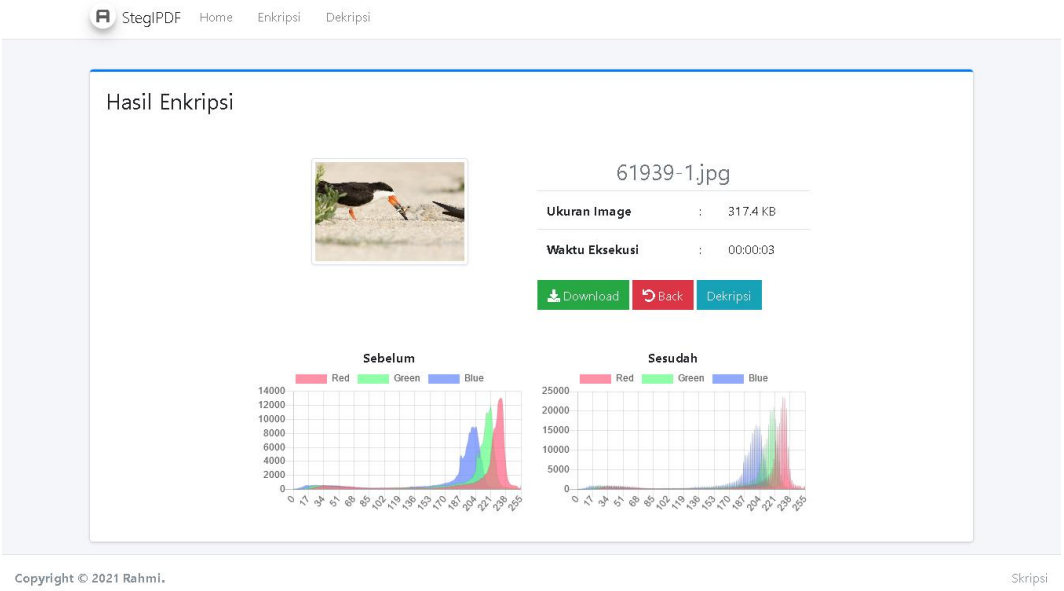
2. Halaman *Menu* Enkripsi



Gambar 4. 13. Halaman *Menu*Enkripsi

Pada halaman menu enkripsiini digunakan untuk mengenkripsi *file pdf* ke *file image*. Langkah pertama yang harus dilakukan adalah dengan meng*input* atau *uploadfilepdf* dan *fileimage*, lalu meng*inputkey* atau *password*. Setelah itu, enkripsi dengan menekan tombol Enkripsi.

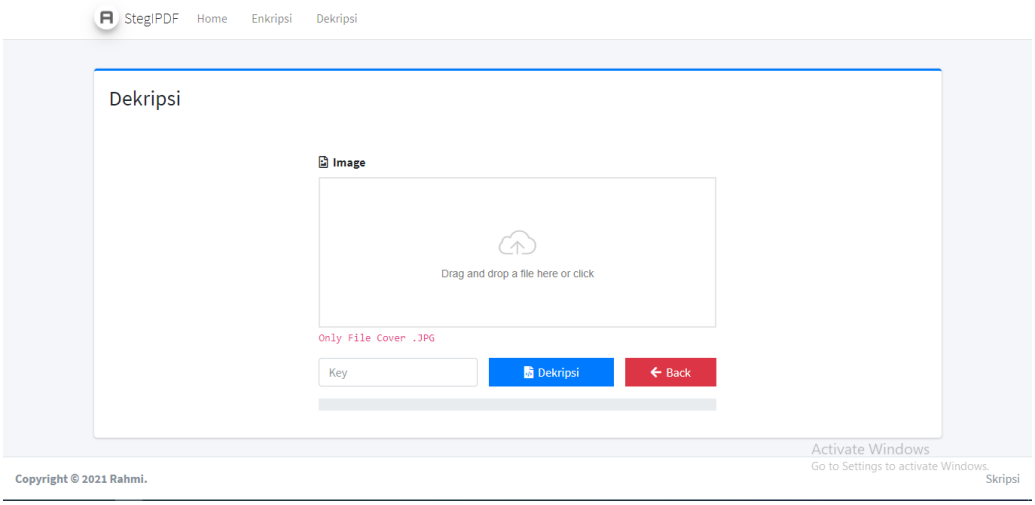
3. Halaman Hasil Enkripsi



Gambar 4. 14. Halaman Hasil Enkripsi

Pada halaman ini digunakan untuk menampilkan hasil dari *file pdf* yang sudah di enkripsi ke dalam *image*. Maka *spectrum* gambar sebelum dan sesudah di enkripsi akan mengalami perubahan dikarenakan gambar awal telah disisipkan file pdf, dan perubahannya tergantung dari hasil *spread spectrum*-nya.

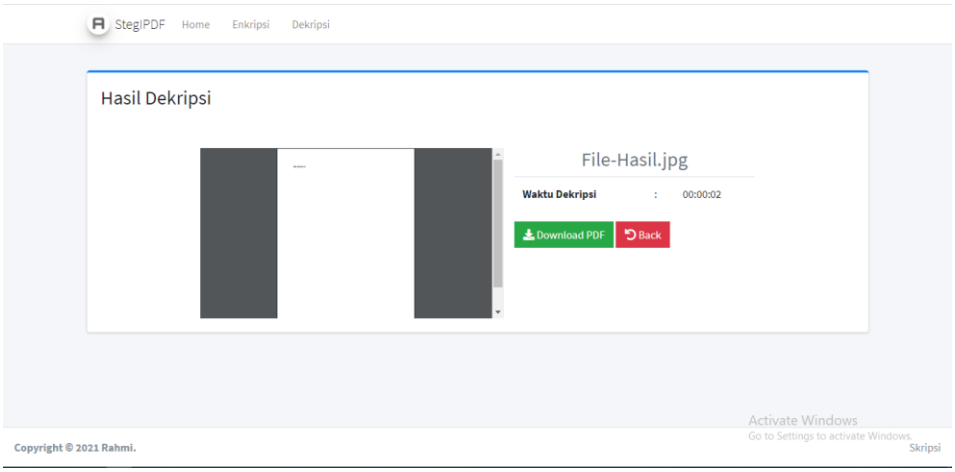
4. Halaman *Menu* Dekripsi



Gambar 4. 15. Halaman *Menu* Dekripsi

Pada halaman menu dekripsi ini digunakan untuk mendekripsi *file cover image*. Langkah pertama yang dilakukan *inputcover image* yang sudah terenkripsi, lalu meng*inputkey* atau *password* sesuai dengan *key* yang telah terenkripsi ke dalam *cover image*, lalu menekan tombol dekripsi untuk memulai proses Dekripsi.

5. Halaman Hasil Dekripsi



Gambar 4. 16. Halaman Hasil Dekripsi

Pada halaman ini digunakan untuk menampilkan hasil dari *file cover image* yang sudah di dekripsi.

D. Pengujian

Pada tahapan pengujian sistem penulis melakukan pengujian terhadap program yang telah dibuat. Pengujian yang dilakukan untuk mengetahui kemampuan aplikasi dalam menangani kondisi yang tidak diinginkan dengan tujuan untuk mempersiapkan banyak hal, selain itu untuk melakukan validasi dalam menerima dan jika dalam *testing* program menemui *error* maka program bisa di perbaiki.

Pengujian aplikasi dilakukan dengan menggunakan metode pengujian yaitu pengujian *blackbox, whitebox* dan tabel hasil pengujian.

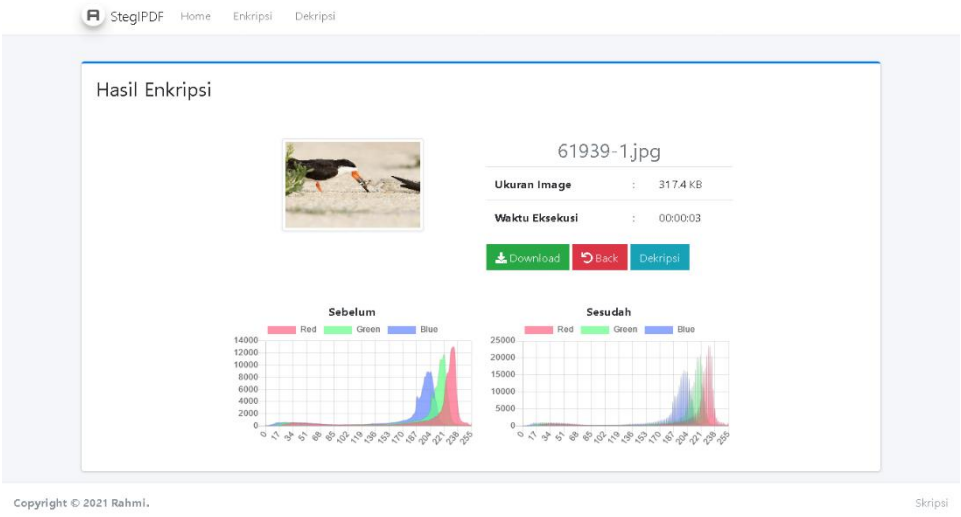
1. Pengujian *Blackbox*

Pengujian *Blackbox* (*blackbox testing*) adalah salah satu metode pengujian yang berfungsi pada sisi fungsionalitas yang ada dalam sistem. Kemudian membandingkan hasil keluaran sistem dengan hasil yang diharapkan. Bila hasil

yang diharapkan sesuai dengan hasil pengujian, artinya aplikasi sesuai dengan desain yang telah ditentukan sebelumnya. Jika belum sesuai maka perlu dilakukan pengecekan lebih lanjut dan perbaikan.

a. *Blackbox* Enkripsi berhasil

Tabel 4. 1. Jika Enkripsi Berhasil

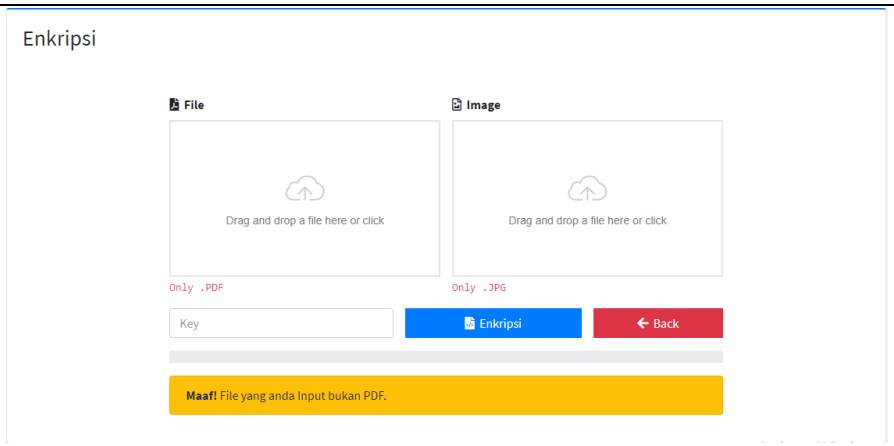
Test Factor	Hasil	Kesimpulan
Jika *pdf* berhasil di enkripsi ke dalam image	✓	Berhasil, karena tidak ada muncul pesan validasi.
Screenshot		
		

Gambar 4. 17. Berhasil Terenkripsi

b. *Blackbox*bukan *file PDF*yang di*input*

Tabel 4. 2. Bukan *file PDF*yang di*input*

Test Factor	Hasil	Kesimpulan
Jika *file pdf* tidak di*input* atau yang di*input* selain *pdf*	✓	Berhasil, karena muncul pesan validasi tentang selain *file pdf* yang ter*input*

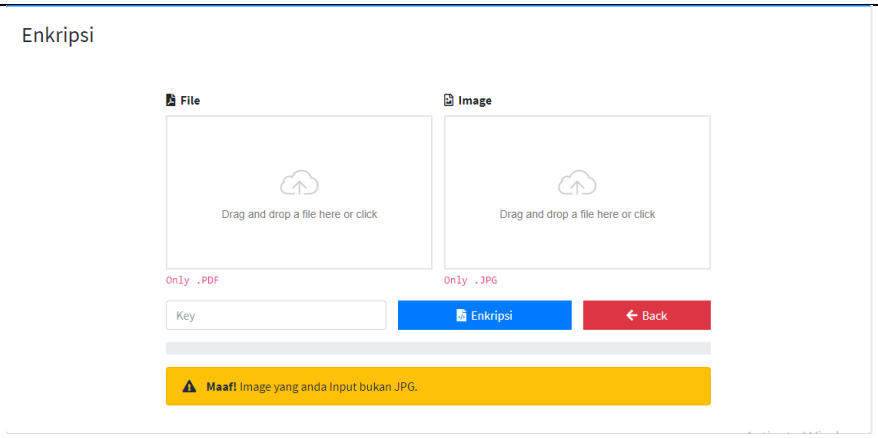
Screenshot


Gambar 4. 18. Bukan *file PDF* yang di*input*

c. *Blackbox* bukan *fileimage jpg*yang di*input*

Tabel 4. 3. Bukan *file image jpg* yang di*input*

Test Factor	Hasil	Kesimpulan
Jika *file image* bukan *jpg*atau selain *file image* yang di*input*.	✓	Berhasil, karena muncul pesan validasi tentang selain *file image jpg*yang ter*input*
Screenshot		

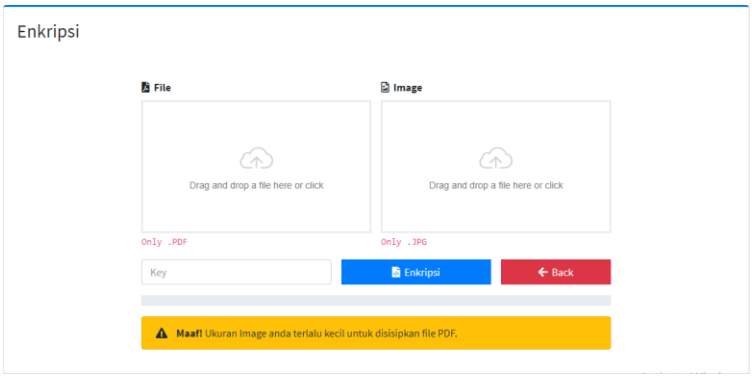


Gambar 4. 19. Bukan *file imageJPG* yang di*input*

d. *Blackbox* gagal terenkripsi

Tabel 4. 4.*PDF* gagal terenkripsi

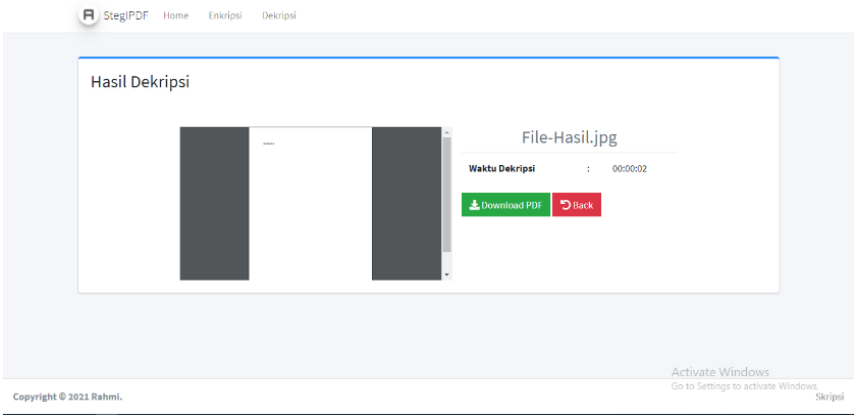
Test Factor	Hasil	Kesimpulan
Jika *file pdf* gagal terenkripsi ke dalam *file image.*	✓	Berhasil, karena muncul pesan validasi tentang gagal terenkripsi
Screenshot		



Gambar 4. 20. Gagal Terenkripsi

e. *Blackbox* Berhasil Terdekripsi

Tabel 4. 5. Berhasil Terdekripsi

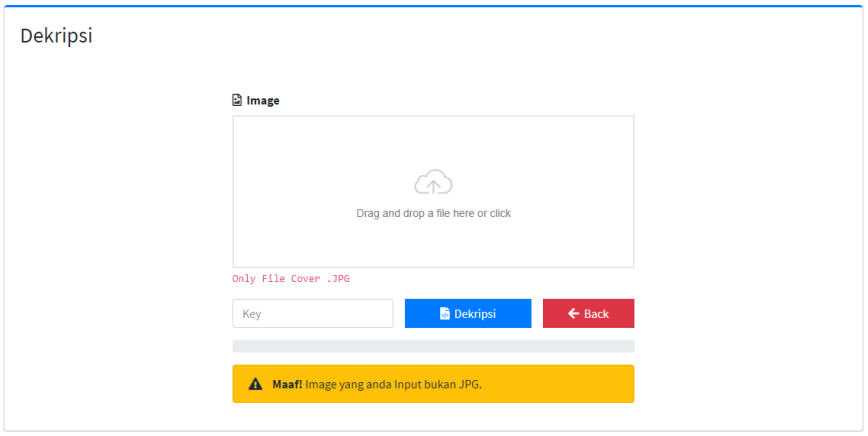
Test Factor	Hasil	Kesimpulan
Jika *file cover image* berhasil terdekripsi	✓	Berhasil, karena muncul pesan validasi tentang berhasil terdekripsi
Screenshot		
		

Gambar 4. 21. Berhasil Terdekripsi

f. *Blackbox* yang di*input* bukan *image cover JPG*

Tabel 4. 6. Bukan *Image Cover JPG*

Test Factor	Hasil	Kesimpulan
Jika bukan *file cover imagejpg* yang di*input*	✓	Berhasil, karena muncul pesan validasi tentang selain *file cover image* bertipe *jpg*
Screenshot		

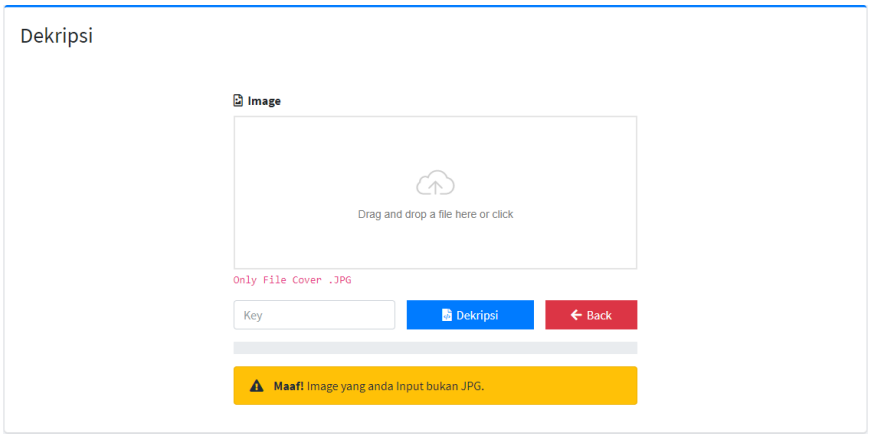


Gambar 4. 22. Bukan *cover image JPG*

g. *Blackboxkey* atau *password* salah

Tabel 4. 7.*Password/Key* Salah

Test Factor	Hasil	Kesimpulan
Jika *password* atau *key*nya salah	✓	Berhasil, karena muncul pesan validasi pesan *password* yang dimasukkan salah
Screenshot		

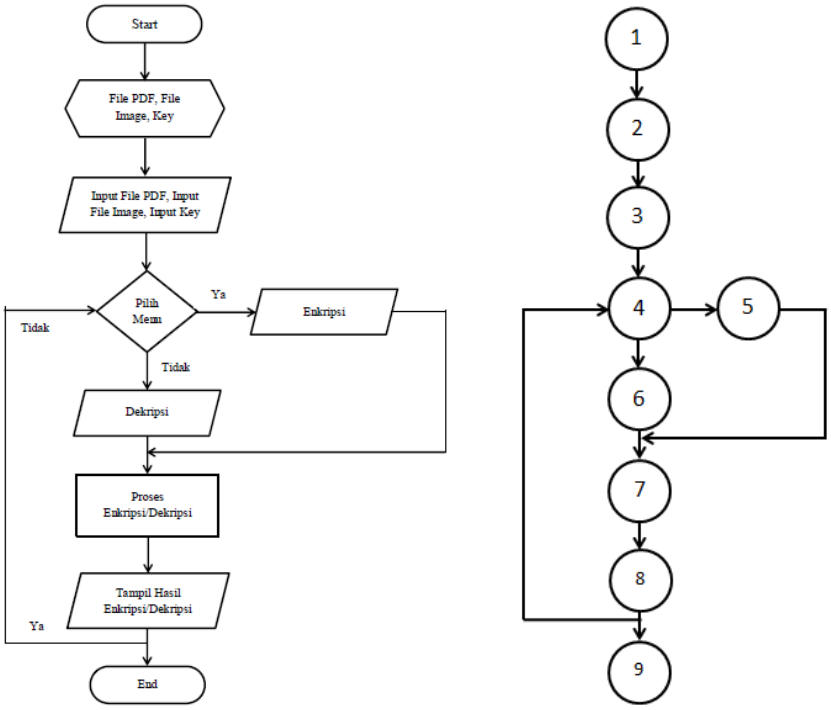


Gambar 4. 23.*Password/Key* Salah

2. Pengujian *Whitebox*

Pada tahap pengujian *White Box,* dilakukan pengujian terhadap pengecekan kode program yang digunakan pada aplikasi ini. Pengujian *White Box* dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan lancar dan tidak terdapat kesalahan dalam logika pemrograman.

a. Halaman Enkripsi



Gambar 4. 24.*Flowchart* dan *Flowgraph* Enkripsi

Dari gambar *flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexity V*(*G*) dari *Edge* dan *Node* :

Dengan Rumus : V (G) = E – N + 2

N(*node*)= 9

E(*edge*)= 10

P(*Predikat node*) = 2

Penyelesaian= V(*G*) = E – N + 2

= 10 – 9+2

= 3

Predikat = P + 1

= 2 + 1

= 3

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *region* = 3

3) Independent *path* pada *flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9

Path 2 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9

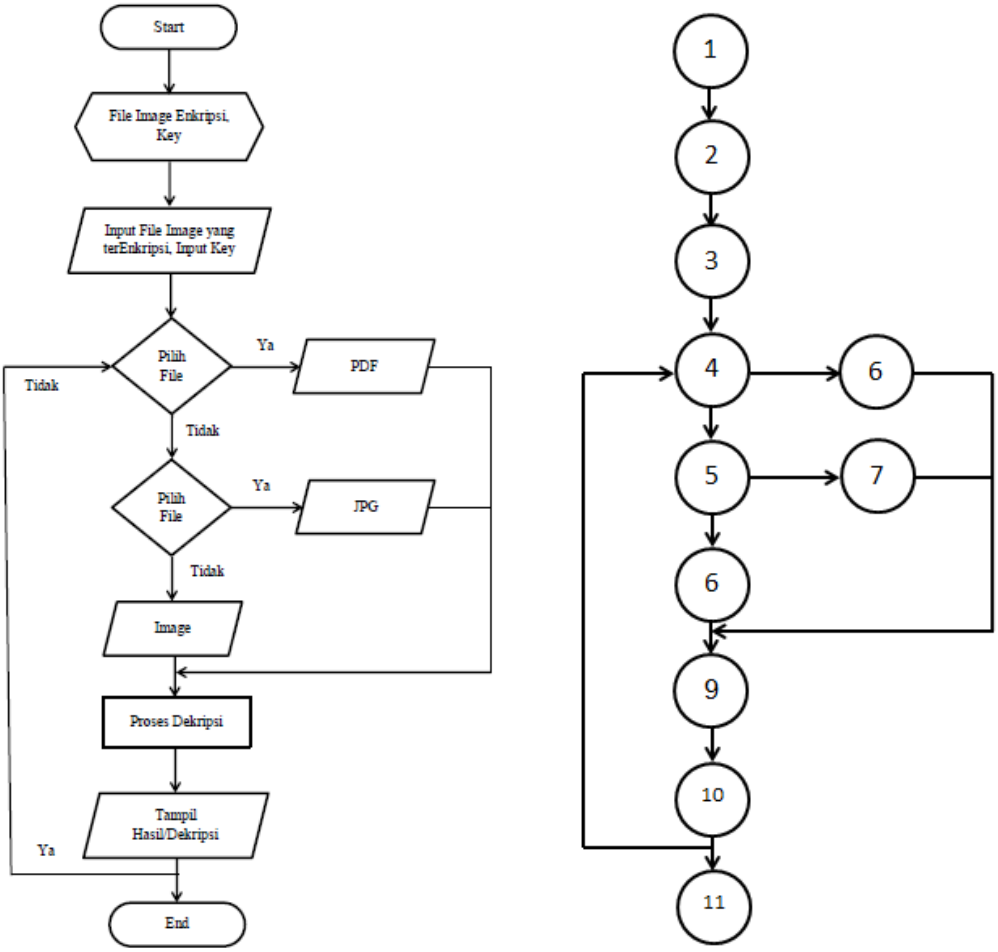
Path 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 4

4) *Grafik Matriks*

Tabel 4. 8. Grafik Matriks Enkripsi

	1	2	3	4	5	6	7	8	9	E – 1
1		1								1 – 1 = 0
2			1							1 – 1 = 0
3				1						1 – 1 = 0
4					1	1				2 – 1 = 1
5							1			1 – 1 = 0
6							1			1 – 1 = 0
7								1		1 – 1 = 0
8				1					1	2 – 1 = 1
9										
SUM (E+1)										1+1 = 2

b. Halaman Dekripsi



Gambar 4. 25.*Flowchart* dan *Flowgraph* Dekripsi

Dari gambar *flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexity V*(*G*) dari *Edge* dan *Node* :

   Dengan Rumus : V (G) = E – N + 2

   N(*node*)= 11

   E(*edge*)= 13

   P(*Predikat node*) = 3

Penyelesaian= V(*G*) = E – N + 2

= 13 – 11+ 2

= 4

Predikat = P + 1

= 3 + 1

= 4

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *region* = 4

3) Independent *path* pada *flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 11

Path 2 = 1 – 2 – 3 – 4 – 7 – 9 – 10 – 11

Path 3 = 1 – 2 – 3 – 4 – 5 – 8 – 9 – 10 – 11

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 4

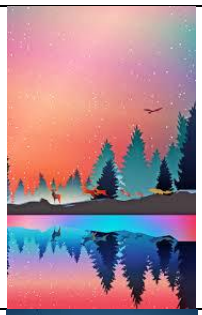
4) *Grafik Matriks*

Tabel 4. 9.*Grafik Matriks* Dekripsi

	1	2	3	4	5	6	7	8	9	10	11	E – 1
1		1										1 – 1 = 0
2			1									1 – 1 = 0
3				1								1 – 1 = 0
4					1	1						2 – 1 = 1
5						1		1				2 – 1 = 1
6									1			1 – 1 = 0
7									1			1 – 1 = 0
8									1			1 – 1 = 0
9										1		1 – 1 = 0
10				1							1	2 – 1 = 1
11												
SUM (E + 1)												1 + 1 + 1 = 3

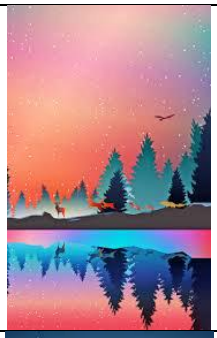
3. Hasil Pengujian

Tabel 4. 10. Hasil Pengujian Enkripsi

Sebelum Enkripsi	Sesudah Enkripsi	Ukuran PDF	Ukuran Gambar Awal – Akhir	Berhasil / Gagal
		2.72 kb	7.39 kb – 147 kb	Berhasil
		4.46 kb	12.5 kb – 147.2 kb	Berhasil
		233 kb	124 kb – 124 kb	Gagal
		385 kb	740 kb – 11700 kb	Berhasil
		278 kb	54.9 kb – 54.9 kb	Gagal

Resolusi dari *cover image* harus lebih besar dari *file pdf* yang ingin disisipkan, apabila resolusi *cover image* lebih kecil maka akan terjadi kegagalan ketika ingin dienkripsi.

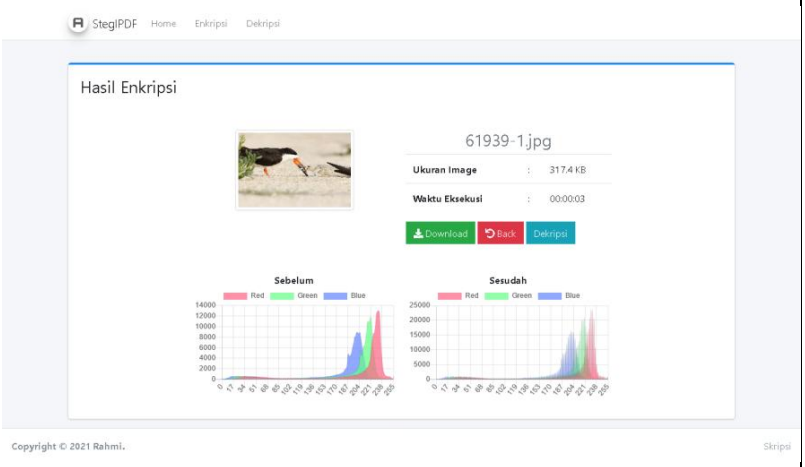
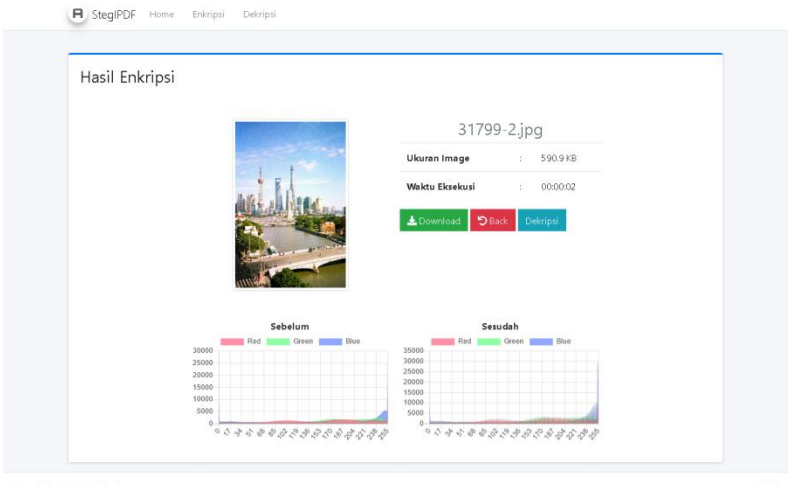
Tabel 4. 11. Hasil Pengujian Dekripsi

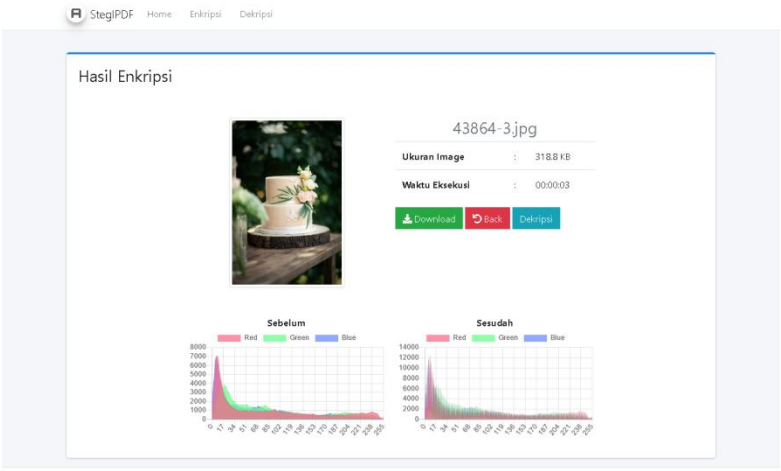
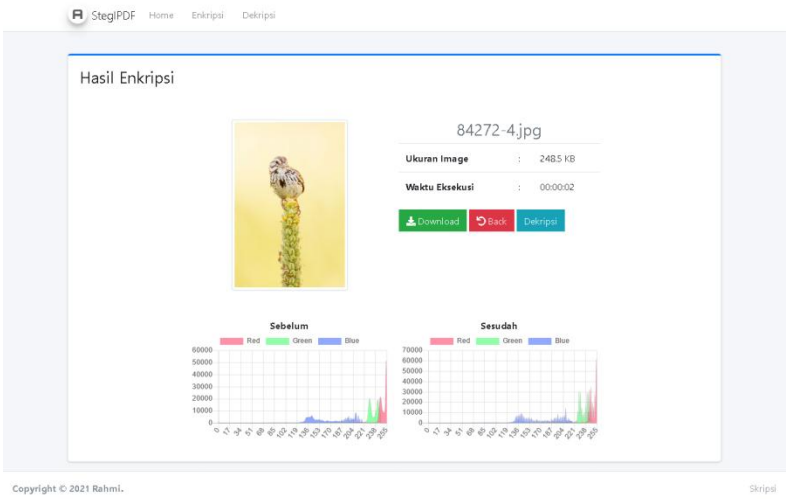
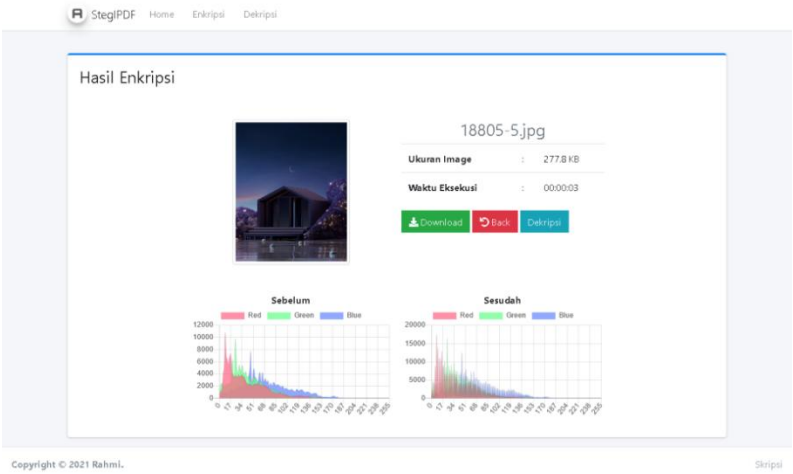
File Cover	Ukuran File Cover	Waktu Eksekusi
	147 kb	1 detik
	147,2 kb	1 detik
	124 kb	0 detik
	11700 kb	17 detik
	54.9 kb	0 detik

Tabel 4. 12. Hasil Pengujian File

Jenis File	Ukuran File	Ukuran File Cover	Waktu Eksekusi	Hasil
Surat	184 kb	131 kb	27 detik	✓
Artikel	1.129 kb	131 kb	23 detik	✓
Gambar	351 kb	525 kb	1 menit 9 detik	✓

Tabel 4. 13. Hasil Spectrum Gambar Enkripsi

File Hasil Enkripsi	Spektrum Gambar Sebelum dan Sesudah
	
	

File Hasil Enkripsi	Spektrum Gambar Sebelum dan Sesudah
	
	
	

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan oleh penulis maka dapat ditarik kesimpulan yaitu :

Pembuatan aplikasi steganografi ini menggunakan metode *spread spectrum* dalam mengenkripsi/dekripsi sebuah pesan rahasia ke dalam gambar berformat *jpg/jpeg*, media yang digunakan yaitu *file* pdf, sistem ini dibangun menggunakan aplikasi *Sublime Text 3* sebagai rancangan aplikasinya, dan untuk tampilan nya menggunakan *web browser.*

File yang telah di enkripsi/dekripsi akan tersimpan secara otomatis di penyimpanan yang tersedia, adapun pengujian sistemnya mendapatkan hasil yang sesuai dengan apa yang telah ditargetkan.

Aplikasi ini dibuat dengan tujuan dapat membantu penggunanya untuk mengirimkan sebuah data-data yang bersifat rahasia, agar pesan dapat sampai ke tangan penerima tanpa menimbulkan kecurigaan pada pihak lain.

Hasil penelitian menunjukkan bahwa enkripsi dan dekripsi dilakukan dengan pengujian tabel sebanyak 5 kali, yang dimana resolusi dari *coverimage* harus lebih besar dari *filepdf* yang ingin disisipkan, apabila resolusi *cover* lebih kecil maka akan gagal ketika di enkripsi. Jika ukuran resolusi *coverimage* terlalu besar maka waktu eksekusinya saat di enkripsi akan berlangsung 1 detik hingga 1 menit. Untuk dekripsi waktu eksekusinya kurang lebih dari 15 detik dibanding

ketika di enkripsi. Dan hasil dari *file pdf* yang sudah di enkripsi ke dalam *image* maka *spectrum* gambar sebelum dan sesudah di enkripsi akan mengalami perubahan dikarenakan gambar awal telah disisipkan *file pdf*, dan perubahannya tergantung dari hasil *spread spectrum*-nya.

B. Saran

Sistem ini memiliki beberapa kekurangan, untuk membangun sebuah sistem yang baik, maka diperlukan beberapa saran sebagai berikut :

1. Aplikasi ini hanya dapat menggunakan *file* gambar berformat jpg/jpeg, maka dari itu penulis berharap kedepannya dapat dikembangkan agar bisa menggunakan semua format gambar yang ada.
2. Penulis berharap agar aplikasi ini nantinya dapat berkembang menjadi aplikasi desktop dan dapat dimanfaatkan dikemudian hari.

DAFTAR PUSTAKA

Achmad Ardiansyah, Mepa Kurniasih. (2018). Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit. *Teknologi Informasi Vol. XIII Nomor 3*, 96-101.

Anshori, Y., Dodu, A. E., & Purwaningsih, M. (2019). Aplikasi Steganografi pada Media Citra Digital Menggunakan Metode Least Significant Bit (LSB). *Sains dan Teknologi Informasi*, *5*(1), 1-10.

Arius Dony. 2008. *Kriptografi Keamanan Data & Komunikasi*. Yogyakarta: Graha Ilmu.

Budi Sutedjo Dharma Oetomo, "*Perencanaan Dan Pembangunan Sistem Informasi*".

Dzul Fahmi Faisal Fadilah, Raufi Baihaqi Syahruzah. (2020). Enkripsi File Gambar Menggunakan Kriptografi Gost. *Karya Ilmiah*, 1-4.

Khasadika Donovan, Ekojono, Imam Fahrur Rozi. (2016). Aplikasi Steganography untuk Enkripsi Image to Image dengan Metode Spread Spectrum. *Informatika Polinema*, 29-33.

Lekso Budi Handoko, Chaerul Umam. (2019). Penyembunyian Pesan Menggunakan Steganografi dengan Metode LSB dan Enkripsi Kriptografi. *Prosiding SENDI_U*, 74-80.

Lestari, T., Nurmaesa, N., & Mariana, A. R. (2017). Aplikasi Steganografi Untuk Menyisipkan Pesan Dalam Media Image. *Jurnal Sisfo Tek Global*, *7*(2).

Mardiansyah, A., & Yusfarizal, Y. (2021). APLIKASI PENYISIPAN PESAN PADA GAMBAR MENGGUNAKAN SPREAD SPECTRUM DAN GHOST BERBASIS ANDROID. *IT (INFORMATIC TECHNIQUE) JOURNAL*, *8*(1), 81-92.

Munir, Rinaldi (2006). *Steganografi dan Watermarking*. URL

Reno, S. (2017). Algoritma Steganografi dengan Metode Spread Spectrum Berbasis PCMK. *MULTINETICS*, *3*(2), 32-37.

Rudiyanto, R. (2018). *implementasi metode spread spectrum dan kriptografi rsa dalam penyembunyian pesan pada citra digital berbasis android* (Doctoral dissertation, Universitas Multimedia Nusantara).

Saidah, S., Ibrahim, N., & Widianto, M. H. (2019). Pengamanan Pesan pada Steganografi Citra dengan Teknik Penyisipan Spread.

Susanto, Agus. 2006. *Studi Implementasi Steganografi pada Berkas MIDI*. Departemen Teknik Informatika: Institut Teknologi Bandung.

Tri Hariyono Reiza Hafidz, T. (2016). Enkripsi Gambar Menggunakan Algoritma Secure Image Protection. *EEPIS Final Project*.

Wandani. Henny. Teknik Steganografi End of File (EOF) dan Rani Public Key Cryptosystem. Universitas Sumatera Utara.

Widianto, S. R. (2017). Algoritma Steganografi dengan Metode Spread Spectrum Berbasis PCMK. *Multinetics Vol 3 No. 2*, 32-37.