APLIKASI PENGAMAN DATA DAN INFORMASI BERLAPIS DENGAN METODE STEGANOGRAFI LSB, KRIPTOGRAFI OPENSSL DAN MD5 BERBASIS WEB

Achmed Robeth Muzaki[1)], Ema Utami[2)]
[1), 2)] Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : achmed.m@students.amikom.ac.id[1)], ema.u@amikom.ac.id[2)]

Abstrak

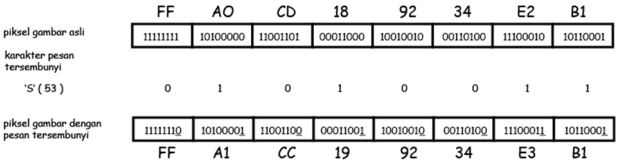
Setiap manusia memiliki data ataupun informasi rahasia, dan cara menyimpannya pun setiap orang berbeda-beda. Di jaman modren ini , mereka biasanya menyimpannya pada media komputer atau prangkat elektronik yang dimilikinya. Ada yang di simpan dalam aplikasi catatan atau aplikasi yang mengharuskan penggunanya untuk login. Aplikasi-aplikasi ini lah yang sering menjadi incaran para pirates untuk diambil data dan informasi rahasianya. Dengan menggunakan 3 teknik sekaligus dalam mengamankan data dan informasi rahasia diyakini dapat membuat tingkat keamannya lebih tinggi. Diantaranya dengan Steganografi dimana data dan informasi rahasia tersebut disembunyikan di media digital. OpenSSL yang notabene adalah sebuah perpustakaan kriptografi dan toolkit SSL. Lapisan terkahir adalah Md5 yang berfungsi sebagai pengengkripsi password. Untuk memudahkan pengguna dalam merahasiakan informasi nya, aplikasi dibuat dalam bentuk web site, dengan begitu pengguna hanya membutuhkan akses internet untuk menyimpan data dan informasi pentingnya ke dalam sebuah media digital. Hasilnya , aplikasi ini bisa berjalan dan bisa melakukan proses enkripsi dan deskripsi pada platform desktop dan smartphone.

Kata kunci: *OpenSSL, Steganografi, Kriptografi.*

1. Pendahuluan

Kebutuhan masyarakat akan keamanan informasi, dengan adanya teknologi informasi, data-data informasi rahasia yang seharusnya tidak boleh diketahui oleh orang lain kecuali pemilik informasinya sangat mungkin terjadi, karena hal tersebut termasuk dalam teknologi informasi dalam hal keamanan informasi. Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan, data, atau informasi dengan cara menyamarkannya menjadi bentuk tersandi yang tidak mempunyai makna.[1] . Dengan membuat algoritma kriptografi berlapis di yakini akan membuat data dan informasi tersebut memiliki level keamanan lebih tinggi.Lapisan itu diantaranya adalah Steganografi LSB,Kriptografi OpenSSL dan Md5.

Steganografi adalah ilmu menyembunyikan teks pada media lain yang telah ada sedemikian sehingga teks yang tersembunyi menyatu dengan media itu. Media tempat penyembunyian pesan tersembunyi dapat berupa media teks, gambar, audio atau video.Sedangkan Steganografi LSB (*Least Significant Bit Insertion*) adalah dengan mengubah nilai Least Significant Bit Insertion pada *byte* intensitas piksel dengan teks yang ingin di sembunyikan. [2]. Misalnya pada gambar gray level piksel direpresentasikan sebagai 1 byte. Jika terdapat 8 piksel bernilai {FF, A0, CD, 18,92, 34, E2, B1} (dalam heksa desimal) dan huruf pada teks ingin disembunyikan adalah "S" yang dengan nilai ASCII "S" adalah 53 (dalam heksimal) atau (01010011). Proses grafi dengan LSB menghasilkan gambar dengan pesan yang tersembunyi dengan deretan piksel bernilai {FE, A1, CC, 19,92, 34, E3, B1}. Gambar.1 mengilustrasikan proses penyisipan ini. Untuk mendapatkan teks tersembunyi kembali cukup dengan membaca LSB tiap piksel.



Gambar 1 *Contoh Steganografi dengan teknik penyisipan bit pada LSB (Rifki Sadikin,2012)*

Selanjutnya dengan kriptografi OpenSSL pada dasarnya adalah dua alat dalam satu: perpustakaan kriptografi dan toolkit SSL. Pada umumnnya OpenSSL bekerja penuh pada Sistem Oprasi Unix dan juga Windows, terutama pada program yang menggunakan bahasa C dan C++, tetapi kita juga bisa menggunakan pada bahasa pemograman seperti Python, Perl dan PHP. OpenSSL pada PHP adalah abstraksi tingkat tinggi dari OpenSSL API. Tidak seperti Perl atau Python[3].
OpenSSL ini merupakan command line tool yang menggunakan berbagai fungsi kriptografi OpenSSL's crypto library dari shell. diantaranya dapat digunakan untuk :

1. Penciptaan RSA, DH dan DSA parameter kunci,
2. Penciptaan sertifikat X.509, CSRs dan CRLs,
3. Perhitungan Pesan Digests ,
4. Enkripsi dan Dekripsi dengan Ciphers ,
5. Pengujian SSL / TLS Client dan Server ,
6. Penanganan S/MIME signed or encrypted mail.

Lapisan terakhir adalah Md5, Bermula dari fungsi hash. Fungsi hash adalah sebuah fungsi yang masukannya adalah sebuah pesan dan keluaran sebuah sidik pesan (*message fingerprint*). Sidik pesan sering juga disebut message digest ,Fungsi hash dapat digunakan untuk mewujudkan beberapa layanan keamanan jaringan misalnya untuk keutuhan data dan otentikasi pesan , konsep dasar fungsi hash banyak dipakai dalam sistem kriptografi MD5 dan SHA[4].
MD5 merupakan fungsi Hash yang sering digunakan untuk mengamankan suatu jaringan komputer dan internet yang sengaja dirancang dengan tujuan sebagai berikut :

1) *Keamanan*: Hal ini tidak bisa dielakkan karena tidak satupun sistem algoritma yang tidak bisa dipecahkan. Serangan yang sering digunakan untuk menjebol algoritma Hash adalah dengan menggunakan serangan brute force.
2) *Kecepatan*: Software yang digunakan mempunyai kecepatan yang tinggi karena didasarkan sekumpulan manipulasi operan 32 bit.
3) *Simple*: Tanpa menggunakan sturktur data yang kompleks

Dari tinjauan pustaka yang dilakukan dalam penelitian ini, didapati aplikasi yang dibuat hanya bisa berjalan pada satu platform, seperti yang dilakukan oleh Alfian Abdul Jalid (2013), mebuat aplikasi pengamanan data dan informasi dengan metode steganografi lsb dan algoritma kriptografi tripledes menggunakan bahasa pemrograman C# [5], tentu saja hanya berjalan pada komputer saja , karena bahasa pemograman C# menghasilkan aplikasi berbasis desktop. Dan juga oleh Ronald Arie Bowo Supardi (2013), melakukan perancangan aplikasi pengiriman pesan rahasia menggunakan steganografi dan kriptografi dengan teknik substitusi berbasis mobile[6], aplikasi yang hasilkan oleh perancangan ini hanya berjalan pada sistem Android.

Berdasarkan uraian diatas, maka penulis mencoba mengembangkan aplikasi berbasis web yang digunakan untuk keamanan data dan informasi. Karena pengguna Internet mencakup pengguna Smartphone dan Komputer. Untuk perancangan aplikasi ini menggunakan UML (*Unified Modeling Language*) merupaka bahasa yang distandarkan sebagai peralatan untuk dokumen analisis dan perancangan suatu sistem perangkat lunak [7]

2. Pembahasan

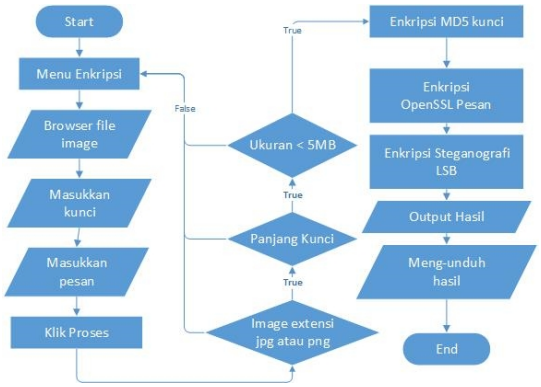
Tahap pertama kali yang penulis lakukan dalam penelitian ini adalah pengumpulan data sehingga bisa digunakan untuk masuk dalam tahap selanjutnya yaitu tahap kedua atau bisa disebut menganalisis data. Hasil dari menganalisis data yang penulis telah lakukan diantaranya kelemahan pada sistem aplikasi yang lama, yaitu:

1. Hanya berjalan pada platform tertentu.
2. Masih menggunakan algoritma standar yang dengan adanya perkembangan teknologi berkemungkinan untuk lebih cepat di bobol / hack.
3. Pengamanannya kurang kuat karena kunci yang digunakan untuk enkripsi tidak di enkripsi lagi.

Dari rujukan data diatas membuat penulis untuk membuat sistem aplikasi baru dengan kelebihan :

1. Bisa berjalan pada semua platform dengan syarat memiliki akses internet dan browser yang mendukung javascript.
2. Menggunakan pustaka OpenSSL yang menjadi bagian 4 jenis kriptografi terbaik pada bahasa pemograman PHP.
3. Pengamanan kunci yang kuat, kunci yang digunakan diencripsi.

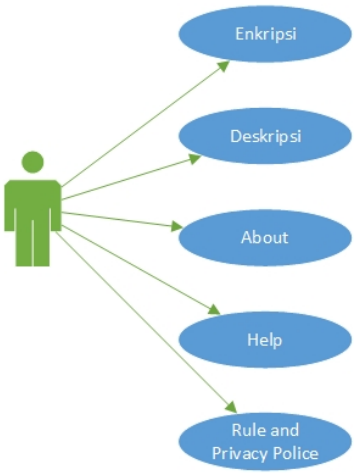
Tahap yang ketiga adalah perancangan, untuk perancangannya menggunakan UML. Perancangan sistem yang akan dilakukan meliputi tiga tahap, yaitu perancangan prosedural, perancangan proses dan perancangan interface / antarmuka.
Perancangan prosedural ini berisi tentang flowchart dari aplikasi ini. Flowchart proses enkripsi bisa dilihat pada gambar 2.



Gambar 2 *Flowchart enkripsi*

Dari gambar 2 dapat dijelaskan, pertama kali sistem akan menampilkan menu enkripsi, menampilkan form yang harus diisi untuk proses enkripsi, diantaranya file image, kunci, dan pesan yang akan di sembunyikan. Selanjutnya sistem akan mengecek ekstensi image, panjang kunci dan ukuran file yang akan diproses. Selanjutnya sistem mengenkripsi kunci dengan md5, yang hasilnya akan digunakan untuk proses enkripsi pesan dengan OpenSSL. Selanjutnya hasil dari enkripsi OpenSSL akan di sembunyikan kedalam file dengan metode Steganografi LSB. Hasil dari proses diatas selanjutnya bisa diunduh dalam bentuk file gambar.

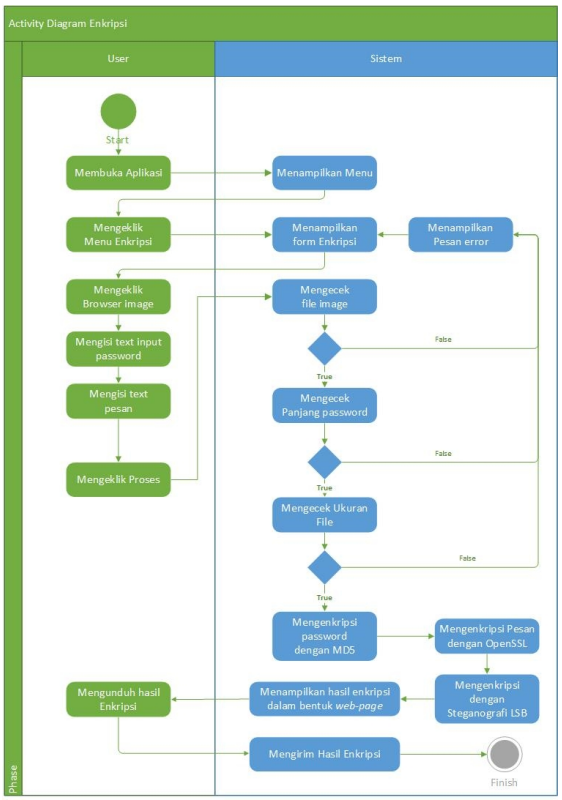
Perancangan proses aplikasi ini dimulai dari *Use Case Diagram, Activity Diagram, Squence Diagram* sampai *Class Diagram*.
Use case diagram merepresentasikan dekripsi lengkap tentang interaksi yang terjadi anatara para actor dengan sistem[8]. *Use case* dari aplikasi yang akan dirancang bisa dilihat di gambar 3.



Gambar 3 *Usecase Diagram*

Dari gambar 3 bisa di jelaskan, sebuah actor yang dapat melakukan interaksi agar system melakukan berbagai proses seperti enkripsi, deskripsi , *about, help* dan *rule and police.*

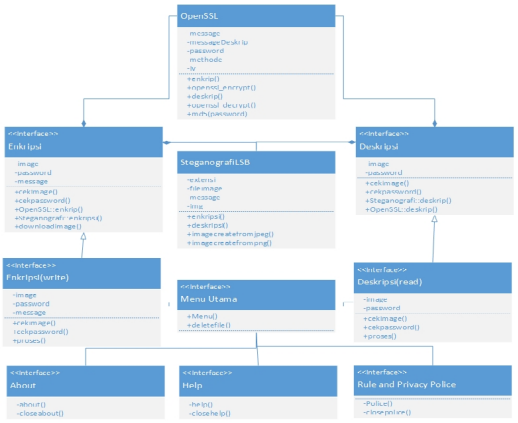
Activity diagram menggambarkan aktifitas-aktifitas, objek, *state, transisi state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas[9]. Gambar 4 menunjukkan aktiviti diagram proses enkripsi.



Gambar 4 *Activity Diagram Enkripsi*

Pada gambar 4 bisa menjelaskan urutan aktifitas user dan sistem mulai dari membuka aplikasi yang dilakukan user sampai berakhir pada sistem dengan mengirim hasil enkripsi.

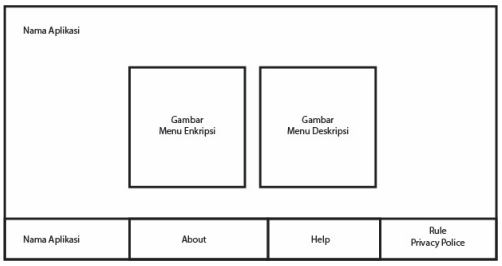
Class diagram adalah mendeskripsikan tipe dari objek dalam sistem dan variasi jenis hubungan yang ada diantara objek tersebut. *Class diagram* juga menunjukkan properti dan operasi dari sebuah class dan kendala yang mempengaruhi bagaimana objek terhubung [10]. *Class diagram* dari aplikasi ini bisa dilihat di gambar 5.



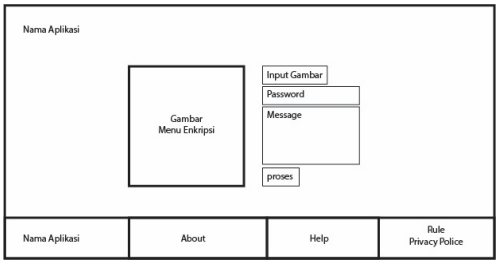
Gambar 5 *Class Diagram*

Dari gambar 5 bisa di jelaskan, hubungan antar class dari class menu utama dengan class-class di sekitarnya yang berisi detail tiap kelas.

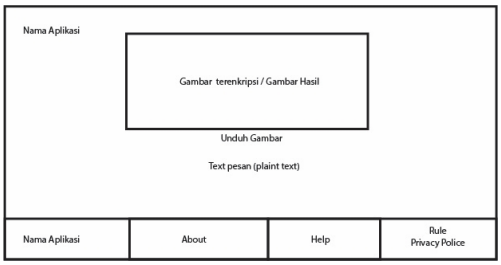
Perancangan *interface* / antarmuka program berperan untuk menghubungkan antara pengguna dan aplikasi. Peranan *interface* sangat penting karena dengan *interface* yang baik akan membuat penggunaan aplikasi menjadi lebih mudah dan tidak membingungkan. Perancangan *interface* didesain dengan konsep *user friendly*. Dari gambar 6,7 dan 8 menunjukkan rancangan tampilan dari aplikasi ini.



Gambar 6 *Perancangan tampilan menu utama pada layar desktop*



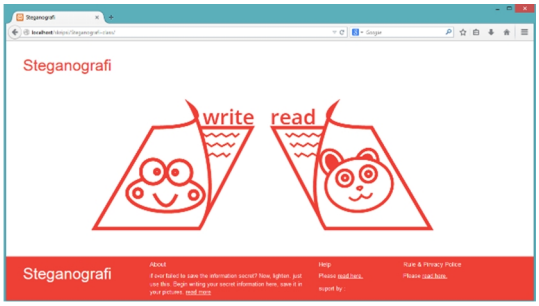
Gambar 7 *Perancangan tampilan menu enkripsi pada layar desktop*



Gambar 8 *Perancangan tampilan hasil proses enkripsi pada layar desktop*

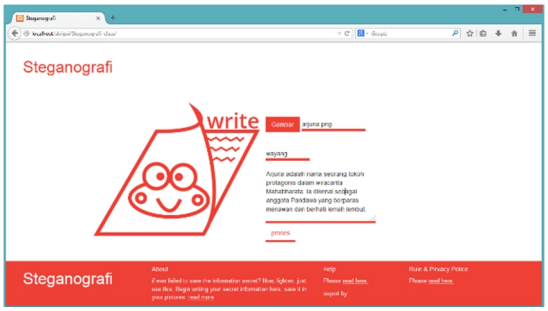
Dari gambar 6 menujukkan rancangan tampilan untuk menu utama, yang berisi bagian area enkripsi, deskripsi *about* , *help* dan *rule police*. Penjelasan selanjutnya untuk gambar 7 berisi form-form yang harus diisi untuk proses enkripsi. Untuk gambar 8 adalah rancangan tampilan hasil dari proses enkripsi , tampilan tersebut berisi gambar hasil dan pesan yang disembunyikan.

Tahap yang keempat adalah implementasi. Tahap ini meliputi penulisan code untuk membangun sebuah aplikasi ini.
Tahap terkahir adalah tahap pengujian. Gambar 9,10 dan 11 adalah hasil pengujian proses enkripsi menggunakan file gambar berekstensi *.jpg dan menggunakan perangkat *desktop* .

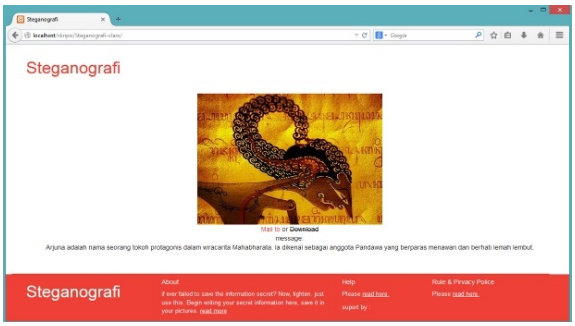


Gambar 9 *Pengujian tampilan menu utama pada layar desktop*

Dari gambar 9 menunjukkan tampilan dari menu utama. Selanjutnya gambar 10 menunjukkan hasil pengujian untuk mengisi form-form untuk proses enkripsi. Gambar 11 menunjukkan tampilan hasil proses enkripsi.

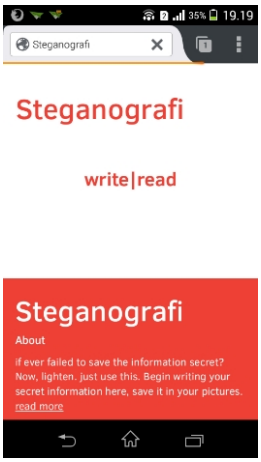


Gambar 10 *Pengujian mengisi form password dan message pada desktop*



Gambar 11 *Pengujian tampilan hasil proses enkripsi pada desktop*

Gambar 12,gambar 13 dan gambar 14 adalah proses deskripsi menggunakan file hasil proses enkripsi yang berekstensi .png dan menggunakan perangkat *smartphone*.



Gambar 12 *Pengujian tampilan menu utama pada layar smartphone*

Gambar 13 P*engujian pengisian form password pada smartphone*

Gambar 14 *Pengujian tampilan hasil proses deskripsi pada smartphone*

Dari gambar 12 menunjukkan gambar pengujian menuutama dengan platform smartphone, selanjutnya gambar 13 menujukkan pegujian pegisian form yang akan di gunakan untuk proses deskripsi. Pada gambar 14 menujukkan hasil dari proses deskripsi.

Berikut adalah hasil-hasil pengujian dari aplikasi yang dibuat dari beberapa tabel. Dari table 1,menunjukkan hasil ujicoba menggunakan *file* dengan ukuran 523Kb dengan kecepatan jaringan 65Mb. Untuk melihat kecepatan proses enkripsi dan deskripsi bisa dilihat pada table 2 dan table 3. Untuk hasil uji dengan jumlah karakter bisa dilihat pada table 4. Sedangkan hasil uji pada perbedaan ukuran file sebelum dan sesudah enkripsi bisa dilihat pada table 5.

Tabel 1. Hasil ujiacoba enkripsi dan deskripsi dengan *platform* yang berbeda

No	Platform	Detail	Status
1	Desktop Leptop	ASUS A44H Series	√
2		Compaq CQ41	√
3	Smarphone	Sony Experia M Dual	√
4		Sony Experia E Dual	√
5		Lenovo A369i	√

Tabel 2. Hasil ujicoba kecepatan enkripsi dan hasil ukuran enkripsi dengan ukuran file yang berbeda

No	Ukuran sebelum proses (Kb)	Ukuran setelah proses (Kb)	Waktu Yang diperlukan (detik)	Status
1	8	4	0.67836308	√
2	56	390	0.18943190	√
3	554	1.889	1.75298500	√
4	1.018	4.351	3.09402489	√
5	2.451	7.488	6.43105912	√
6	3.423	11.456	9.59213519	√

Tabel 3. Hasil ujicoba kecepatan deskripsi dan hasil ukuran enkripsi dengan ukuran file yang berbeda

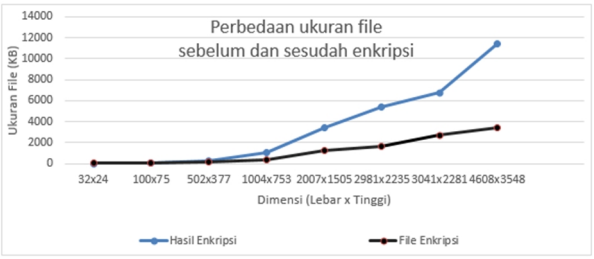
No	Ukuran sebelum proses (Kb)	Waktu Yang diperlukan (detik)	Status
1	4	0. 0022568	√
2	390	0.00278115	√
3	1.889	0.02204799	√
4	4.351	0.15424299	√
5	7.488	0.21365809	√
6	11.456	0.41423511	√

Tabel 4. Hasil Ujicoba dengan jumlah karakter yang berbeda

No	Jumlah karakter	Dimensi : 2981x2235 Ukuran : 1643KB	
		Hasil (KB)	Waktu
1	1	5406	4.340
2	100	5407	4.322
3	250	5407	4.331
4	500	5408	4.350
5	1000	5411	4.351
6	2000	5415	4.357

Tabel 5. Perbedaan ukuran file sebelum dan sesudah enkripsi

No	Dimensi	Ukuran sebelum proses (KB)	Ukuran sebelum proses (Kb)	Status
1	32x24	33	2	√
2	100x75	35	12	√
3	502x377	112	288	√
4	1004x753	328	1000	√
5	2007x1505	1199	3399	√
6	2981x2235	1643	5406	√
7	3041x2281	2711	6735	√
8	4608x3548	3423	11456	√



Gambar 15 *Grafik perbedaan ukuran file sebelum dan sesudah enkripsi*

Ringkasan penjelasan dari hasil tabel-tabel pengujian aplikasi di atas adalah:

1. Pada tabel 1 bisa di jelaskan,dengan menggunakan *platform* yang berbeda tidak terlalu mempempengaruhi kecepatan dimana kecepatan

bandwith yang digunakan sama yaitu 65Mbps karena proses enkripsi dan deskripsi terletak pada web server.

2. Pada tabel 2,menunjukkan perbedaan kecepatan enkripsi berbanding lurus dengan besarnya ukuran file yang dienkripsi. Begitu juga pada table 3 dengan kecepatan saat proses deskripsi. Semakin besar *file* yang akan di proses maka semakin lama prosesnya.
3. Pada table 4 menunjukkan pengujian dengan jumlah panjangnya karakter, hanya mempengaruhi sedikit perubahan pada ukuran *file* hasil.
4. Pada tabel 5 menunjukkan ukuran file sebelum dan sesudah enkripsi juga berbanding lurus, dimana besarnya ukuran file sebelum enkripsi akan menghasilkan *file* yang lebih besar dari ukuran *file* sebelumnya.

3. Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya maka dapat diambil kesimpulan yaitu :

1. Penelitian ini menghasilkan aplikasi berbasis web yang mampu menyembunyikan data dan informasi rahasia yang dirancang dengan menggunakan UML (*Unified Modeling Language*) .
2. Aplikasi ini dapat melakukan proses enkripsi dan dekripsi pada platform *desktop* dan *smartphone*.
3. Semakin besar ukuran file yang dienkripsi maka semakin lama prosesnya dan semakin besar ukuran file yang dihasilkan.
4. Aplikasi ini dibuat melalui tahap analisis yaitu dengan menggunakan analisis kebutuhan dan analisis kelayakan, setelah itu tahap perancangan yaitu dimulai rancangan aplikasi dan rancangan interface serta implementasi dan pengujian sistem atau aplikasi.

Dalam pembuatan suatu aplikasi tentunya terdapat kekurangan seperti halnya aplikasi pengamanan data informasi menggunakan metode steganografi LSB, kriptografi OpenSSL dan MD5 ini. Agar aplikasi ini dapat lebih sempurna, ada beberapa saran dari penulis yang bisa menjadi pertimbangan agar aplikasi pengamanan data dan informasi dengan cara enkripsi dan dekripsi ini menjadi lebih baik, diantaranya yaitu :

1. Tampilan aplikasi bisa menjadi daya tarik khusus, sehingga penambahan theme atau tema pada aplikasi sehingga aplikasi tidak membosankan.
2. Ekstensi file gambar yang dapat dienkripsi lebih banyak karena dalam penelitian ini *file* yang dienkripsi hanya teks dalam ekstensi .jpg , .jpeg dan .png.
3. Maksimal *file* gambar yang dapat dienkripsi saat ini maksimal 5MB sehingga bisa ditambah lagi untuk kapasitasnya.
4. File gambar yang dihasilkan bisa mencapai 3 kali lipat dari *file* aslinya, untuk saran penelitian selanjutnya, bisa mempertimbangkan *file* yang dihasilkan untuk tidak terlalu bersar.
5. Data dan informasi yang di sembunyikan tidak hanya berupa teks, tetapi bisa menyembunyikan data dan informasi berupa *file* berektensi seperti misalnya : .txt , .docx dan .xlsx untuk Microsoft Office , dan .odt dan dan .odf untuk Liber Office.
6. Diharapkan untuk penelitian selanjutnya *user* dapat lebih leluasa memilih algoritma apa yang digunakan dalam proses pengamanan data dan informasi rahasianya

Daftar Pustaka

[1] Munir,R.*Kriptografi* . Bandung:Informatika. 2006.
[2] Rifki Sadikin, *Pengantar Kriptografi dan Keamanan Jaringan*, Yogyakarta: Andi Offset,2012.
[3] Pravir Chandra, Matt Messier , John Viega, *Network Security with OpenSSL*, Sebastopol: O'Reilly, 2002.
[4] Rifki Sadikin, *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: Andi Offset , 2012.
[5] Alfian Abdul Jalid . *Aplikasi Pengamanan Data dan Informasi dengan Metode Steganografi Lsb dan Algoritma Kriptografi Tripledes Menggunakan Bahasa Pemrograman C#.* Yogyakarya: Teknik Informatika STMIK AMIKOM Yogayakarta. 2013
[6] Ronald Arie Bowo Supardi. *Perancangan Aplikasi Pengiriman Pesan Rahasia Menggunakan Steganografi dan Kriptografi dengan Teknik Substitusi Berbasis Mobile*. Yogyakarya: Teknik Informatika STMIK AMIKOM Yogayakarta. 2013
[7] Kenneth E. Kendall dan Julie E. Kendall. *Analsis dan Perancangan Sistem (Systems Analysis and Design*. Edisi ke-5. Jilid 2. Jakarta : PT. Indeks Kelompok Gramedia. 2003.
[8] Nugroho, Adi. *Rekayasa Perangkat Lunak mwnggunakn UML dan JAVA.* Yogyakarta: Andi Offset, 2009.
[9] Haviluddin *,"Memahami Penggunaan Uml (Unified Modelling Language)"* Jurnal Informatika Mulawarman Vol 6 No. 1 Febuari 2011.
[10] Fowler, Martin. *UML Distilled*. Edisi ke-3. Terjemahan Tim Penerjemah Penerbit Andi Offset. Yogyakarta: Andi Offset. 2004

Biodata Penulis

Achmed Robeth Muzaki, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2015. Saat ini menjadi Startup dan Freelancer Web Developer dan Game Developer.

Dr. Ema Utami, S.Si, M.Kom, memperoleh gelar Sarjana Sains (S,Si) dari Ilmu Komputer Universitas Gadjah Mada pada tahun 1997, gelar Magister Komputer (M.Kom) dari Ilmu Komputer Universitas Gadjah Mada pada tahun 2002, dan gelar Doktor (Dr) dari Ilmu Komputer Universitas Gadjah Mada pada tahun 2010. Sejak tahun 1998 menjadi dosen di STMIK AMIKOM Yogyakarta dan saat menjadi Wakil Direktur I Bidang Akademik Program Pascasarjana STMIK AMIKOM Yogyakarta.