Jurnal Tera
Volume 1, Issue 2, September 2021
Page 185-201

E-ISSN : 2776-9666
P-ISSN : 2776-1789

APLIKASI STEGANOGRAFI PADA MEDIA GAMBAR MENGGUNAKAN *ALGORITMA LEAST SIGNIFICAN BIT*

Fakhria Nur Sabrina[1]
[1] Jurusan Teknik Informatika, Fakultas Teknik dan Informatika, Universitas Dian Nusantara, Jakarta

Corresponding author
E-mail: fakhria.nur.sabrina@undira.ac.id

Diterima : 15/03/2021
Direvisi : 25/04/2021
Dipublikasi : 19/05/2021

Abstrak: Keamanan data merupakan hal yang terpenting untuk pertukaran informasi, banyaknya kejahatan dalam teknologi informasi yang terjadi akhir – akhir ini membuat orang harus berhati – hati terhadap data atau informasi yang akan dikirim. Untuk menghindari adanya kejahatan itu dibutuhkan suatu aplikasi yang bisa menjamin keamanan data yang akan dikirimkan. Informasi atau data yang akan dikirimkan bisa disembunyikan dalam suatu cover objek atau bisa disamarkan dengan mengenkripsi data tersebut. Sebuah pesan dapat dikatakan aman apabila pesan tersebut telah sampai kepada penerima, dalam keadaan pesan tidak dimanipulasi atau diubah selama pengiriman. Berbagai macam teknik yang digunakan untuk menyembunyikan pesan dengan steganografi atau kriptografi. Disini pesan rahasia disembunyikan menggunakan teknik steganografi, yaitu pesan rahasia akan disisipkan dalam sebuah cover objek berupa gambar, dan hanya orang yang berhak yang bisa membuka pesan rahasia tersebut. Perangkat lunak ini dibangun pada perangkat mobile berbasis Android, berdasarkan hasil implementasi dan pengujian yang dilakukan, aplikasi ini dapat mengenkripsi dan menyisipkan pesan kedalam cover objek. Tidak ada perbedaan antara citra asli dengan hasil citra yang dihasilkan oleh aplikasi ini, sehingga keamanan data yang dikirimkan dengan menggunakan aplikasi ini bisa terjamin.

Kata Kunci: Steganografi, Least Significant Bit, Citra, Android

PENDAHULUAN

Internet merupakan media komunikasi yang sudah popular dan sangat berkembang pesat. Kemudahan dalam penggunaan dan fasilitas yang lengkap merupakan keunggulan umum yang dimiliki oleh internet. Penggunaan internet yang sudah bersifat global tanpa mengenal ruang, waktu dan birokrasi, dimana akses data dan informasi dapat melampaui batas – batas negara dan protokoler. Di samping itu, pelaku bisnis juga mempergunakan internet sebagai

media untuk pengiriman data penting. Kebutuhan informasi yang tersedia di internet dengan jangkauan global, perlu dikembangkan pengamanan pada pesan, agar hanya beberapa orang saja yang dapat mengakses atau mendapatkan pesan tersebut.

Seiring dengan perkembangan teknologi saat ini, penggunaan internet tidak hanya di gunakan pada desktop komputer atau laptop saja, akan tetapi sudah meluas ke teknologi mobile. Penggunaan intenet menggunakan teknologi mobile sudah meluas ke masyarakat. Salah satu alasan masyarakat memilih teknologi mobile adalah untuk kemudahan mencatat segala informasi yang mereka anggap penting. Di padu dengan teknologi internet, teknologi mobile kini semakin di nikmati. Hal ini membuat internet menjadi semakin popular bahkan tampak seperti kebutuhan wajib bagi masyarakat modern. Kemudahan dalam mengakses internet kini seperti berada dalam gengaman setiap orang. Dengan kemudahan tersebut, bertambah pula kejahatan dalam sistem informasi. Dengan berbagai teknik pengambilan informasi secara ilegal yang berkembang, Banyak yang mencoba untuk mengakses informasi yang bukan haknya. Sejalan dengan berkembangnya teknologi mobile yang sangat cepat, harus juga di ikuti dengan berkembangnya pengamanan sistem informasi pada mobile yang bisa melindungi data.

Berbagai macam metode yang digunakan untuk melindungi data yang dirahasiakan dari orang yang tidak berhak. Banyak dilakukan upaya dalam mengamankan suatu data penting dengan menggunakan sistem kriptografi yang melakukan enkripsi sebelum data penting tersebut di transmisikan. Tindakan pengamanan tersebut ternyata belum cukup dalam mengamankan suatu data karena adanya peningkatan komputasi. Berbeda dengan kriptograpi, steganografi menyembunyikan pesan rahasia agar orang awam tidak menyadari keberadaan dari pesan yang disembunyikan. Teknik ini sering digunakan untuk menghindari kecurigaan orang dan menghindari keinginan orang untuk mengetahui isi pesan rahasia tersebut. Banyaknya kejahatan dalam teknologi informasi yang terjadi akhir – akhir ini, membuat orang harus hati – hati terhadap data atau informasi yang dikirim. Untuk menghindari adanya kejahatan pada data yang akan dikirim oleh suatu instansi atau perorangan, data atau informasi yang akan dikirim harus dapat disembunyikan atau di ubah menjadi sebuah kode atau file yang lainnya. Itu digunakan agar orang yang berhak untuk menerima pesan tersebut yang dapat membaca pesan tersebut.

Pesan dikatakan telah aman jika pesan yang dikirim bisa sampai dengan utuh ke tangan penerima, artinya pesan tidak diubah atau dimanipulasi selama pengiriman oleh pihak ketiga. Disisi penerima pesan, ia tentu ingin memastikan bahwa pesan yang ia terima adalah pesan yang masih asli, bukan pesan yang ditambah – tambah atau dikurangi. Pesan yang dikirim tanpa perlindungan atau keamanan dapat disalahgunakan oleh orang yang tidak bertanggung jawab. Hal tersebut dapat membuat kerugian bagi pemilik data atau informasi tersebut. Agar terhindar dari kerugian tersebut diperlukan keamanan pada data yang akan dikirim tersebut. Agar tidak disalahgunakan oleh orang yang tidak bertanggung jawab.

Berdasarkan permasalahan diatas, perlu dirancang sebuah sistem yang mampu menyembunyikan data, melindungi data, dan memberikan keamanan terhadap data yang akan di kirimkan agar bisa diakses oleh orang yang berhak saja. Dengan aplikasi steganografi data yang dikirim akan disembunyikan dan di lindungi sehingga data tersebut aman dari penyalahgunaan orang yang tidak bertanggung jawab. Dan dengan aplikasi steganografi pesan yang dikirim hanya bisa diterima dan dibaca oleh orang yang berhak saja. Untuk itu, dalam penelitian ini penulis ingin mengambil judul "PERANCANGAN APLIKASI PESAN RAHASIA PADA DIGITAL CITRA MENGGUNAKAN

STEGANOGRAFI PADA PERANGKAT MOBILE BERBASIS ANDROID". Aplikasi ini diharapkan dapat bisa memberikan keamanan dalam mengolah informasi yang akan digunakan.

KAJIAN PUSTAKA

Steganografi

Steganografi berasal dari bahasa Yunani yaitu *steganos* yang berarti tersembunyi/menyembunyikan dan *graphy* yang artinya tulisan, sehingga secara keseluruhan artinya adalah tulisan yang disembunyikan. Steganografi telah digunakan sejak sekitar 2500 tahun yang lalu untuk kepentingan politik, militer, diplomatik, serta untuk kepentingan pribadi sebagai alat. Catatan pertama tentang steganografi ditulis oleh sejarahwan Yunani, Herodotus, yaitu ketika Histiacus dipenjarakan oleh Raja Darius di Susa pada abad 5 SM. Histiacus harus mengirim pesan rahasia kepada Aristagoras, dengan cara mentato pesan pada kulit kepada seorang budak dan ketika rambut budak itu mulai tumbuh, Histiacus mengutus budak itu ke Militus untuk mengirim pesan di kulit kepalanya tersebut kepada Aristagoras. Cara tersebut dilakukan untuk menghindari isi pesan diketahui oleh pihak yang tidak diinginkan di tengah perjalanannya.

Definisi informal pertama tentang steganografi diformulasikan oleh Simmons dalam "*The Prisoners' Problem and The Subliminal Channel* ". Dua narapidana, Alice dan Bob berada dalam ruangan sel yang berbeda di bawah pengawasan sipir penjaga, Eve. Mereka saling berkomunikasi berencana untuk melarikan diri, namun sipir penjaga hanya akan mengijinkan komunikasi mereka jika semua komunikasi melalui sipir penjaga. Apabila pesan yang disampaikan Alice kepada Bob tidak mencurigakan, maka pesan akan diteruskan. Dengan kondisi demikian Alice berusaha menyamarkan berita yang disampaikan kepada Bob sebagaimana pesan biasa yang tidak mencurigakan, namun sebenarnya di dalamnya mengandung pesan rahasia yang hanya diketahui oleh mereka berdua.

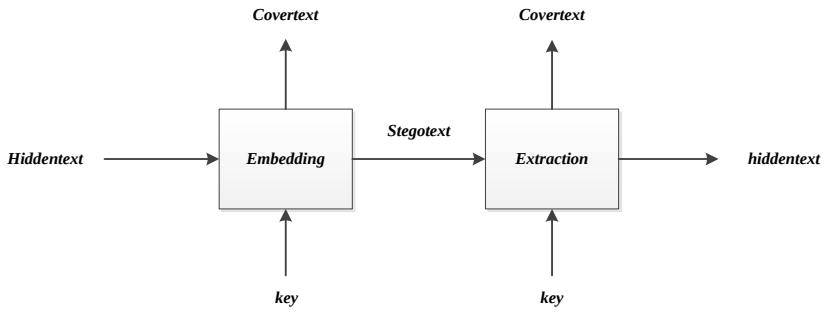
Dalam perkembangannya dengan menggunakan media eletronik dan jaringan komunikasi, pesan bisa dipertukarkan dengan cepat dan mudah tanpa dibatasi jarak dan waktu. Namun kemudahan pertukaran pesan melalui media elektronik mempunyai beberapa resiko, diantaranya resiko penyadapan, pengubahan, dan perusakan pesan, sehingga diperlukan suatu cara yang bisa mengurangi dampak negatif atas terjadinya resiko tersebut. Karena alasan tersebut, muncul penyandian terhadap pesan dengan enkripsi dan deskripsi. Enkripsi (*enciphering*, standar nama menurut ISO 7498-2) dilakukan pada pesan yang akan dikirim dengan cara mengubah pesan asli ke dalam bentuk lain yang sulit untuk dimengerti. Sedangkan deskripsi (*dechiphering*, standar nama menurut ISO 7498-2) dilakukan pada pesan hasil enkripsi yang diterima dengan cara mengubahnya kembali ke bentuk aslinya dengan kunci yang memang diketahui sebelumnya.

Dengan teknik tersebut, pesan yang dikirimkan selalu dalam bentuk yang sulit dimengerti sehingga mengetahui maksudnya hanyalah pengirimnya dan penerima yang dituju. Ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya disebut kriptografi. Semula penyandian sederhana telah efektif untuk menjaga kerahasiaan pesan, tetapi lalu muncul kriptanalisis. Kriptanalisis adalah ilmu dan seni untuk memecahkan pesan terenkripsi menjadi bentuk aslinya tanpa kunci yang diberikan. Dampak penggunaan teknik kriptografi yang cukup mencolok adalah suatu informasi yang tidak terbaca / dipahami, tidak lazim sebagaimana mestinya. Hal tersebut

akan memberikan suatu simpulan bagi pihak yang tidak berwenang melakukan proses kriptanalasis. Karena telah dikenali oleh penyadap, resiko diketahui, diubah, dan dirusaknya pesan asli menjadi lebih besar. Menyadari hal tersebut, para ahli kriptografi mencoba mencari cara baru yang lebih efektif dengan melakukan penyembunyian pesan ke dalam pesan lain sehingga pesan tersebut tidak disadari keberadaannya. Cara ini disebut steganografi yaitu seni atau ilmu yang mengkomunikasikan sebuah pesan dengan suatu cara, tanpa bisa dideteksi oleh pihak lawan.

Berbeda dengan kriptografi yang menjaga kerahasiaan pesan dengan cara mengubah bentuk pesan agar tidak dapat dipahami oleh orang lain, steganografi merupakan suatu teknik penyembunyian pesan pada suatu *medium*. Perlu diperhatikan dalam steganografi, suatu pesan tidak harus berubah, tetapi pesan tersebut disembunyikan pada suatu *medium* agar pesan tersebut tidak terlihat. Salah satu keuntungan steganografi dibandingkan kriptograpi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung pesan tidak menimbulkan kecurigaan bagi pihak ketiga. Steganografi banyak digunakan untuk memberikan tanda (seperti *copyright*) pada suatu karya cipta yang menandakan bahwa karya cipta tersebut bukan bajakan. Selain itu steganografi juga seringkali digunakan untuk menyembunyikan pesan rahasia yang ditujukan kepada orang tertentu.

Meskipun mempunyai maksud yang sama untuk mengamankan informasi namun terdapat perbedaan mendasar antara kriptografi dan steganografi. Kriptografi menyembunyikan / mengolah isi (*content*) pesan sehingga pesan tidak dapat terbaca. Sedangkan steganografi menyembunyikan keberadaan (*existence*) pesan sehingga tidak menimbulkan kecurigaan (*conspicuous*) akan adanya pesan. Steganografi memerlukan media untuk menyembunyikan pesannya, bisa berupa teks, gambar, audio, maupun video.



Gambar 1. Skema Metode Steganografi
Sumber: (Rinaldi Munir, 2006:307)

Pesan yang akan disembunyikan sering diistilahkan sebagai embedded message (*hidenttext*), datanya bisa berupa file, gambar, teks, audio, video, dan lain – lain. Data yang dijadikan media untuk menyembunyikan pesan disebut *cover medium / object (covertext)*. *Cover medium* yang telah ditambahkan pesan rahasia dengan steganografi disebut *stego*-data / object (*stego*text). Sedangkan kunci yang digunakan pada proses penyembunyian maupun rekontruksinya disebut stego-*key*. Pada keadaan yang ideal, siapapun yang melakukan *scan* terhadap data tersebut tidak akan mengetahui bahwa data tersebut mengandung data lain yang rahasia sehingga pengambilan data hanya dapat dilakukan oleh penerima yang berhak.

Blok utama yang membangun sebuah algoritma steganografi adalah pemilihan *cover medium*, algoritma *embedding* dan *extracting*, serta manajemen *stego key*. Secara matematis skema steganografi sudah dapat ditentukan. Anggap Ks adalah *stego key* dari susunan K, semua

kunci rahasia *stego*. M adalah susunan semua pesan yang dapat ditempelkan, dan C adalah susunan semua *cover medium*. Skema steganografik dibentuk oleh dua pemetaan, pemetaan penempelan, Emb dan pemetaan ekstrasi, Ext :
Emb : C x K x M. → C
Ext : C → $M._1$
Sehingga Ext(Emmbc(c,K_s,**m**)) = **m** untuk semua **c** Ɛ C, K_s Ɛ K , and **m Ɛ** M. $s = Emb(c,Ks,$ **m**) disebut *stego work*

Banyaknya teknik dalam steganografi menyebabkan diperlukan adanya pengelompokan atas jenis – jenisnya. Pengelompokan ini diharapkan akan memudahkan pengguna steganografi untuk memilih teknik yang sesuai dan pembuat steganografi untuk mengembangkan teknik – teknik baru dengan lebih terarah. Steganografi sendiri dibedakan menjadi dua jenis menurut tujuannya, yaitu steganografi untuk menghindari deteksi (*data hiding*) dan steganografi untuk menghindari penghapusan data (*document marking*). Jenis pertama kemudian lebih sering disebut sebagai steganografi itu sendiri. Jenis kedua dibagi lagi menjadi dua yaitu *watermaking* dan *fingerprinting*. Manfaat utama dari *watermaking* adalah untuk identifikasi dan menyertakan potongan informasi yang unik pada suatu media tanda disadari orang lain. Kedua hal tersebut umumnya ditujukan untuk menandakan keaslian dari suatu karya yang pada akhirnya bisa meminimalkan tindak pembajakan atas karya tersebut. Steganografi dapat digunakan juga untuk melakukan perawatan atas kerahasiaan informasi yang berharga dari kemungkinan sabotase, pencuri, atau dari pihak yang tidak berwenang. Sayangnya, steganografi juga dapat digunakan untuk alasan yang ilegal. Sebagai contoh, steganografi dapat digunakan oleh para teroris untuk menyamarkan komunikasi mereka dari pihak luar.

Least Significant Bit
Cara paling umum untuk menyembunyikan pesan adalah dengan memanfaatkan *Least Significant Bit* (LSB). Walaupun terdapat kekurangan pada metode ini, tetapi kemudahan implementasinya membuat metode ini tetap digunakan sampai sekarang. Contoh ilustrasinya sebagai berikut : jika digunakan *image* 24 *bit* warna sebagai media, sebuah *bit* dari masing – masing komponen *Red*, *Green*, dan *Blue*, dapat digunakan sehingga 3 *bit* dapat disimpan pada setiap piksel. Sebuah *image* 800 x 600 piksel dapat digunakan untuk menyembunyikan 1.440.000 *bit* (180.000 bytes) data rahasia. Misalnya, di bawah ini terdapat 3 piksel dari *image* 24 *bit* warna :
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
Jika diinginkan untuk menyembunyikan karakter A (100000011) dihasilkan :
(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)
Dapat dilihat bahwa hanya 3 *bit* saya yang perlu diubah untuk menyembunyikan karakter A ini.

Jika pesan = 10 *bit*, maka jumlah byte yang digunakan = 10 byte.
Contoh susunan byte yang lebih panjang :
00110011 10100010 11100010 10101011 00100110
10010110 11001001 11111001 10001000 10100011
Pesan = 1110010111
Hasil penyisipan pada *bit* LSB :
00110011 10100011 11100011 10101010 00100110

10010111 11001000 11111001 10001001 10100011

Pada metode LSB, ukuran data yang akan disembunyikan bergantung pada ukuran *cover-object.*

Perubahan pada LSB ini akan terlalu kecil untuk dideteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Proses ekstraksi pesan dapat dengan mudah dilakukan dengan mengekstrak LSB dari masing – masing piksel pada *stego* secara berurutan dan menuliskannya ke *output file* yang akan berisi pesan tersebut. Keuntungan metode LSB adalah mudah dalam pengimplementasian dan proses *encoding* yang cepat. Pada perkembangannya metode steganografi LSB selain diterapkan pada media *image*, juga bisa diterapkan pada media audio.

Kriteria steganografi yang baik meliputi tiga hal yaitu :

1) *Imperceptible / Undetectability*
   Keberadaaan pesan tidak dapat dipersepsi.
2) *Fidelity*
   Mutu *cover-object* tidak jauh berubah akibat *embedded.*
3) *Recovery*
   Data yang disembunyikan harus dapat diungkapkan kembali.

Kriteria *robustness* tidak terlalu penting karena tujuan yang utama dari steganografi adalah untuk menghindari kecurigaan (lawan tidak menyadari keberadaan pesan tersembunyi).

Sedangkan efektifitas teknik steganografi bisa diukur menggunakan tiga prinsip berikut :

1) Besarnya data, semakin besar / banyak data yang disembunyikan berarti semakin baik tekniknya.
2) Tingkat kesulitan terdeteksi, berhubungan dengan seberapa mudah seseorang dapat mendeteksi adanya pesan yang disembunyikan. Biasanya ada hubungan secara langsung antara besarnya data yang dapat disembunyikan dan seberapa mudah seseorang dapat mendeteksinya. Semakin besar jumlah informasi yang bisa disembunyikan pada suatu media, semakin meningkatkan kesempatan seseorang untuk dapat mendeteksi bahwa ada informasi yang tersembunyi dalam media tersebut.
3) Tingkat kesulitan penghapusan, melibatkan prinsip – prinsip bahwa seseorang yang sedang menyadap media semestinya tidak dapat dengan mudah menghapus data yang disembunyikan.

Suatu hal esensial yang menjadi kelebihan steganografi adalah kemampuannya untuk menipu persepsi manusia, manusia tidak memiliki insting untuk mencurigai adanya arsip – arsip yang memiliki informasi yang tersembunyi di dalamnya, terutama bila arsip tersebut tampak seperti arsip normal lainnya. Namun steganografi juga bukanlah pengamanan yang sempurna karena metode pendeteksi pesan dalam steganografi juga banyak dikembangkan yang disebut staganalisis, yaitu suatu teknik yang digunakan untuk mendeteksi penggunaan steganografi pada suatu data/media. Seorang steganalis tidak berusaha untuk melakukan deskripsi terhadap informasi yang tersembunyi dalam suatu data/media, yang dilakukan adalah berusaha untuk menemukannya. Terdapat beberapa cara yang dapat digunakan untuk mendeteksi steganografi seperti melakukan pengamanatan terhadap suatu data / media dan membandingkannya dengan salinan data/media yang dianggap belum direkayasa, atau berusaha mendengarkan dan membandingkan perbedaaanya denga data/media lain bila data/media tersebut adalah dalam bentuk audio.

Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli *Android* Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel / *smartphone*. Kemudian untuk mengambangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-*Mobile*, dan Nokia. Pada saat perilisan perdana *Android*, 5 november 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode – kode *Android* di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi *Android*. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar – benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). Sekitar September 2007 Google mengenalkan *Nexus One*, salah satu jenis *smartphone* yang menggunakan *Android* sebagai sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporation* dan tersedia di pasaran pada 5 januari 2010. Pada 9 desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android* ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, *Android*, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. sejak *Android* dirilis telah dilakukan berbagai pembaharuan berupa perbaikan *bug* dan penambahan fitur baru.

Pada masa saat ini sebagian besar *vendor – vendor smartphone* sudah memproduksi *smartphone* berbasis *Android*, *vendor – vendor* itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, IMO, Asus, dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi *Android*. Hal ini, karena *Android* itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun. Tidak hanya menjadi sistem operasi di *smartphone*, saat ini *Android* menjadi pesaing utama dari Apple pada sistem operasi Tablet PC. Pesatnya pertumbuhan *Android* selain faktor yang disebutkan di atas adalah karena *Android* itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, Aplikasi dan *Tool* pengembangan, Market aplikasi *Android* serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga *Android* terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

Android dipuji sebagai "*platform mobile* pertama yang lengkap, terbuka, dan bebas".

- Lengkap (*Complete Platform*) : para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform Android*. *Android* merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
- Terbuka (*Open Source Platform*) : *Platform Android* disediakan melalui *lisensi open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. *Android* sendiri menggunakan Linux Kernel 2.6.
- Free (*Free Platform*) : *Android* adalah *platform* / aplikasi yang bebas untuk *develop*. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform Android*. Tidak

ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk *Android* dapat didistribusikan dan diperdagangkan dalam bentuk apa pun.

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari *Android* dilisensikan dibawah GNU, General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah *terms*. *Android* didistribusikan dibawah Lisensi Apache Software (ASL/ Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *open source*, sebaliknya pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka dibawah lisensi apapun yang mereka inginkan. Pengembang aplikasi *Android* diperbolehkan untuk mendistribusikan aplikasi mereka dibawah skema lisensi apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis *Android*. Sebagian besar pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi *Android*. *Eclipse* adalah IDE yang paling popular untuk pengembangan *Android*, karena memmiliki *Android plug-in* yang tersedia untuk memfasilitasi pengembangan *Android*. Selain itu, *Eclipse* juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembang aplikasi *Android*, ini terbukti dengan adanya penambahan *plugins* untuk *Eclipse* untuk membuat projekt *Android* dimana *source software* berlangsung dari situs resminya Google. Akan tetapi, hal diatas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti *Neatbeans* untuk melakukan pengembangan *Android*.(Safaat, 2012:1)

ADT (Android Development Tools)

Andoid Development Tools (ADT) adalah plugin yang didesain untuk IDE *Eclipse* yang memberikan kemudahan dalam mengembangkan aplikasi *Android* dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan kita dalam membuat aplikasi project *Android*, membuat GUI aplikasi, dan menambahkan komponen – komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan *Android* SDK melalui *Eclipse*. Dengan ADT juga dapat melakukan pembuatan *package Android* (.apk) yang digunakan untuk distribusi aplikasi *Android* yang kita rancang. Mengembangkan aplikasi *Android* dengan menggunakan ADT di *Eclipse* sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi *Android*.

UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar - standar 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 perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan perangkat lunak dalam bahasa – bahasa berorientasi objek seperti C++, JAVA, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi *procedural* 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 UML *syntax* mendefinisikan bagaimana bentuk – bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya. Grady Booch OOD (*Object-Oriented Design*), Jin Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).(Yasin, 2012:260)

RAD (Rapid Application Development)

Rapid Application Development (RAD) adalah sebuah model proses perkembangan software sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan adaptasi "kecepatan tinggi" dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan kontruksi berbasis komponen. Jika kebutuhan dipahami dengan baik. Proses RAD memungkinkan tim pengembangan menciptakan "sistem fungsional yang utuh" dalam periode waktu yang sangat pendek (kira – kira 60 sampai 90 hari). Karena dipakai terutama pada aplikasi sistem kontruksi, pendekatan RAD melingkupi *fase – fase* sebagai berikut:

- *Bussiness modeling*
  Aliran informasi di antara fungsi – fungsi bisnis dimodelkan dengan suatu cara untuk menjawab pertanyaan – pertanyaan berikut : informasi apa yang mengendalikan proses bisnis? Informasi apa yang dimunculkan ? siapa yang memunculkannya? ke mana informasi itu pergi? Siapa yang memprosesnya?.
- *Data Modeling*
  Aliran informasi yang didefinisikan sebagai bagian dari *fase bussiness modeling* disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karateristik (disebut artibut) masing – masing objek diidentifikasi dan hubungan antara objek – objek tersebut didefinisikan.
- *Process Modeling*
  Aliran informasi yang didefinisikan di dalam fase *data modeling* ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.
- *Application Modeling*
  RAD mengasumsikan pemakaian teknik generasi keempat. Selain menciptakan perangkat lunak dengan menggunakan bahasa pemograman generasi ketiga yang konvensional, RAD lebih banyak memproses kerja untuk memakai lagi komponen program yang ada (pada saat memungkinkan) atau menciptkan komponen yang bisa dipakai lagi (bila perlu). Pada semua kasus, alat – alat bantu otomatis dipakai untuk memfasilitasi kontruksi perangkat lunak.
- *Testing and turnover*
  Karena proses RAD menekankan pada pemakaian kembali, banyak komponen program telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus diuji dan semua *interface* harus dilatih secara penuh.

Metode Pengujian

Pengujian adalah sebuah proses terhadap aplikasi/program untuk menemukan segala kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan sesuai dengan spesifikasi perangkat lunak yang telah ditentukan sebelum aplikasi tersebut diserahkan kepada pelanggan. Pengujian merupakan proses eksekusi program yang telah selesai dibuat yang bertujuan untuk menemukan kesalahan. Pengujian yang baik adalah pengujian yang dilakukan dengan probabilitas penemuan kesalahan yang tidak diduga, sedangkan pengujian

yang sukses adalah pengujian yang berhasil mengatasi penyelesaian penemuan kesalahan yang tidak diduga. Ada beberapa jenis strategi dalam penguian perangkat lunak yang semuanya memiliki satu tujuan yang sama, yaitu meningkatkan kepercayaan diri pengembang perangkat lunak terhadap fungsi-fungsi perangkat lunaknya. Beranjak dari tujuan ini, suatu perangkat lunak dapat diuji untuk menerima berbagai perlakuan. (Janner Simarmata, 2010:323-324).

Prinsip dasar yang menuntun pengujian perangkat lunak, yaitu:

1) Semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan. Artinya, pengujianmengungkap kesalahan dari kecacatan yang menyebabkan program gagal.
2) Pengujian harus sudah sejak lama dirancanakan sebelum pengujian itu dimulai. Artinya, semua pengujian dapat direncanakan dan dirancang sebelum semua kode dijalankan.
3) Prinsip pareto berlaku untuk pengujian perangkat lunak. Artinya, dari 80% kesalahan yang ditemukan selama pengujian, penelusuran dari semua modul program mencapai 20%.
4) Pengujian harus dimulai dari "pengujian yang kecil" dan berkembang ke "pengujian yang besar". Selagi pengujian berlangsung maju, pengujian mengubah focus dalam usaha menemukan kesalahan pada kluster modul yang terintegrasi dan akhirnya pada sistem.
5) Pengujian yang mendalam tidak mungkin dilakukan karena tidak mungkin untuk mengeksekusi setiap kombinasi jalur skema pengujian yang jumlah jalur permutasi untuk program menengahnya pun besar.(Janner Simarmata, 2010:301-302)

Metode Black Box

"Black Box Testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar" (Soetam Rizky, 2011:264).

Beberapa keuntungan yang diperoleh dari jenis testing ini antara lain:

1) Anggota tim tester tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2) Kesalahan dari perangkat lunak ataupun bug seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
3) Hasil dari *black box* testing dapat memperjelas kontradiksi ataupun keracunan yang mungkin timbul dari eksekusi sebuah perangkat lunak.
4) Proses *testing* dapat dilakukan lebih cepat dibandingkan *white box testing*.

Beberapa teknik *testing* yang tergolong dalam tipe ini antara lain :

1) Equivalence Partitioning
Pada teknik ini, tia inputan data dikelompokan ke dalam grup tertentu, yang kemudian dibandingkan outputnya.

2) Boundary Value Analysis
Merupakan teknik yang sangat umun digunakan pada saat awal sebuah perangkat lunak selesai dikerjakan.Pada teknik ini, dilakukan inputan yang melebihi dari batasan sebuah data, jika perangkat lunak berhasil mengatasi inputan yang salah, maka dapat dikatakan teknik ini telah selesai dilakukan.

3) Cause Effect Graph
Dalam teknik ini, dilakukan proses testing yang menghubungkan sebab dari sebuah inputan dan akibatnya pada output yang dihasilkan.

4) Random Data Selection

Teknik berusaha melakukan proses inputan data yang menggunakan nilai acak. Dari hasil inputan tersebut kemudian dibuat sebuah tabel yang menyatakan validitas dari *output* yang dihasilkan.

5) *Feature Test*
Pada teknik ini dilakukan proses testing terhadap spesifikasi dari perangkat lunak yang telah selesai dikerjakan.

METODE PENELITIAN

Metode pengembangan system dalam penelitian ini menggunakan metode RAD (Rapid Application Development) dengan beberapa tahapan sebagai berikut :

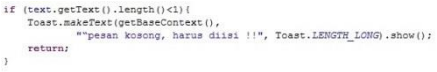
1) Pemodelan Bisnis
Pada tahapan ini penulis menganalisa masalah dan menganalisa kebutuhan dari aplikasi steganografi ini untuk memecahkan masalah.
2) Pemodelan Data
Tahapan ini adalah tahapan definisi mengenai data yang digunakan dalam aplikasi steganografi.
3) Pemodelan Proses
Tahapan ini merupakan tahapan untuk memodelkan data yang sudah didefinisikan dan merancang informasi data sesuai dengan kebutuhan perangkat lunak
4) Pembuatan Aplikasi
Pada tahapan ini membangun aplikasi pesan rahasia steganografi yang menggunakan metode least significant bit yang berbasis sistem operasi Android menggunakan bahasa java.
5) Pengujian dan Pergantian
Pada tahapan ini melakukan pengujian aplikasi sampai berhasil dan menghasilkan program sesuai dengan yang prosedur yang telah dibuat.

HASIL DAN PEMBAHASAN

Pengujian White Box

Pengujian dilakukan dengan menguji kode aplikasi dengan menyediakan test case. Berikut hasil pengujian white-boxnya.

Tabel 1. Encoded Text

	Pengguna harus memasukkan teks, kunci sepanjang 16 karakter, citra, dan nama file. Jika ada yang tidak terpenuhi, maka akan muncul pesan kesalahan dan proses dihentikan.  Pesan kesalahan jika teks kosong :  Pesan kesalahan jika nama file belum diisi :


```
if (fileP.getText().length()<1){
    Toast.makeText(getBaseContext(),
            "nama file kosong, harus diisi !", Toast.LENGTH_LONG).show();
    return;
}
```

Pesan kesalahan jika kunci belum diisi :

```
if (password.getText().length()<16){
    Toast.makeText(getBaseContext(),
            "kunci kosong / kurang dari 16 char, harus diisi !!", Toast.LENGTH_LONG).show();
}
```

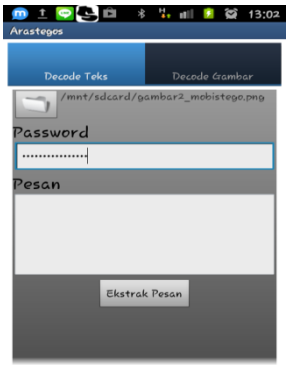
Pesan kesalahan jika image belum di pilih:

```
if (sourceBitmap==null){
    Toast.makeText(getBaseContext(),
            "Gambar belum ada, silahkan input gambar!", Toast.LENGTH_LONG).show();
    return;

}
```

Jika semua parameter dimasukkan dengan benar, proses akan dilakukan citra stego, pesan, key dan direktori penyimpanan stego image ditampilkan

```
AlertDialog.Builder builder = new AlertDialog.Builder(context);
File f = new File(destPath);
String exists=f.exists()?"File dibuat":"File tidak ada";
builder.setMessage(destPath+" "+exists+" – "
+successSaved+" – "+context.getString(R.string.saved)).
setCancelable(false).setPositiveButton(context.getText
        (R.string.ok), new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface arg0, int arg1) {
        // TODO Auto-generated method stub
        Encode.this.finish();
```

Tabel 2. Decoded Text



Pengguna harus memasukkan kunci sepanjang 16 karakter dan citra stego. Jika ada yang tidak terpenuhi, maka akan muncul pesan kesalahan dan proses akan dihentikan.

```
byte[] b = null;
try
{
    int[] pixels = new int[bmp.getWidth() * bmp.getHeight()];
    bmp.getPixels(pixels, 0, bmp.getWidth(), 0, 0, bmp.getWidth(), bmp.getHeight());
    b = LSB.convertArray(pixels);
}
```

Pesan kesalahan jika kunci kurang dari 16 karakter :

```
}
if (password.getText().length()<=16){
    Toast.makeText(getBaseContext(),
            "Kunci < 16 karakter, input dengan benar kunci!",
            Toast.LENGTH_LONG).show();
}
```

Pesan kesalahan jika citra stego tidak diinput :

```
if(sourceBitmap == null) {
    Toast.makeText(getBaseContext(),
            "citra stego tidak ada, silahkan pilih gambar!!",
            Toast.LENGTH_SHORT).show();
    return;
}
```

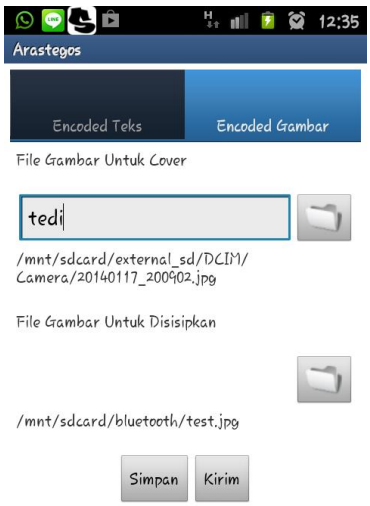
Pesan kesalahan jika tidak ada pesan yang terekstrak :

```
if(decodedMsg == null || decodedMsg.length() < 1) {
    Toast.makeText(getBaseContext(), "Tidak ada pesan yang diekstrak",
            Toast.LENGTH_SHORT).show();
    return;
```

Jika semua parameter dimasukkan dengan benar, proses akan dilakukan dan hasil ekstraksi akan ditampilkan :

```
    txPesan.setText(decodedMsg);
    Toast.makeText(getBaseContext(), "Pesan telah berhasil diekstrak",
            Toast.LENGTH_SHORT).show();
}
}
```

Tabel 3. Encoded Gambar



Pengguna harus memasukkan nama file, image untuk cover, image untuk secret pesan. Jika ada yang tidak terpenuhi maka akan muncul pesan kesalahan dan proses akan dihentikan.

```
}catch(Exception ex){
    Toast.makeText(getBaseContext(), "Error : "+ex.toString()+" error end",
            Toast.LENGTH_SHORT).show();
}
```

Pesan kesalahan jika nama file belum diisi :

```
if (path.getText().length()<1){
    Toast.makeText(getBaseContext(),
            "nama file harus diisi!!!",Toast.LENGTH_LONG ).show();
    return;
}
```

Pesan kesalahan jika image untuk cover image belum di input :

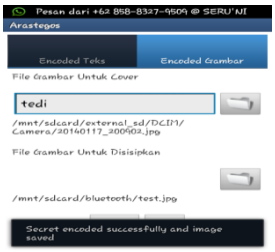
```
if(sourceImage == null) {
    Toast.makeText(getBaseContext(), "Pilih Gambar yang akan menjadi cover",
            Toast.LENGTH_SHORT).show();
    return;
}
```

Pesan kesalahan jika image untuk secret pesan belum di input :

```
if(sourceImage2 == null) {
    Toast.makeText(getBaseContext(), "pilih Gambar yang akan disembunyikan",
            Toast.LENGTH_SHORT).show();
    return;
}
```

Pesan kesalahan jika secret image lebih besar dari cover image :

```
if (newImage ==null) {
    Toast.makeText(getBaseContext(),
            "Image secret tidak dapat dimasukkan ke image " +
            "lain karena ukuran terlalu besar",
            Toast.LENGTH_SHORT).show();
```



Jika semua parameter dimasukkan dengan benar, proses embedding akan dilakukan dan akan ditampilkan pesan :

```
Toast.makeText(getBaseContext(),
        "Secret encoded successfully and image saved",
        Toast.LENGTH_LONG).show();
```

Tabel 4. Decoded Gambar

Pengguna harus memasukkan citra stego yang akan diekstrak, jika ada yang tidak terpenuhi, akan muncul pesan kesalahan dan proses akan dihentikan :

```
} catch (Exception e) {
    // TODO Auto-generated catch block
    Toast.makeText(getBaseContext(), "Error : "+e.toString(),
            Toast.LENGTH_LONG).show();
}
```

Pesan kesalahan jika citra stego tidak diinputkan :

```
if(sourceBitmap == null) {
    Toast.makeText(getBaseContext(), "Pilih Gambar yang akan di ekstrak",
            Toast.LENGTH_LONG).show();
    return;
}
```

Jika semua parameter dimasukkan dengan benar, proses ekstrak akan dilakukan dan akan tampil pesan :

```
imageView2.refreshDrawableState();
Toast.makeText(getBaseContext(), "ekstraksi berhasil dilakukan",
        Toast.LENGTH_LONG).show();
```

Perbandingan Citra Asli dan Citra Steganografi

Perbandingan citra asli dengan citra hasil steganografi (citra stego) dapat dilihat pada table berikut :

Tabel 5. Tabel Perbandingan Citra Asli Dengan Citra Stego

No	Nama Citra Asli	Citra Asli	Nama Citra Stego	Citra Stego
1	oki_setiana_dewi_dan_ory_vitrio-20140112-032-agus.jpg		gambar1_mobistego.png	
2	CYMERA_20131224_112102.jpg		gambar2_mobistego.png	
3	2013-06-10 09.11.10.jpg		gambar3_mobistego.png	
4	photo_5.jpg		gambar4_mobistego.png	
5	CYMERA_20131224_112038.jpg		gambar5_mobistego.png	

Dari tabel diatas, tidak tampak perbedaan yang signifikan dari citra asli dengan citra stego. Perbedaan diantara keduanya dapat dilihat dari ukuran size citra asli dan citra stego

Tabel 6. Tabel Perbedaan Ukuran Size Citra Asli Dan Citra Stego

No	Nama File	Resolusi		Ukuran File		Perubahan Ukuran
		Citra asli	Citra stego	Citra asli	Citra stego	
1	gambar1_mobistego.png	900 x 1325	900 x 1325	581 KB	2.19 MB	3.85x
2	gambar2_mobistego.png	1024 x 768	1024 x 768	453 KB	1.10 MB	2.48x
3	gambar3_mobistego.png	1280 x 856	1280 x 856	0.98 MB	1.75 MB	1.82x
4	gambar4_mobistego.png	640 x 640	640 x 640	526 KB	759 KB	1.44x
5	gambar5_mobistego.png	1024 x 768	1024 x 768	447 KB	1.05 MB	2.40x

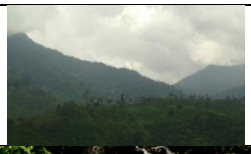
Dari tabel diatas terlihat bahwa tidak ada perubahan pada resolusi citra. Namun terlihat jelas perubahan pada ukuran citra. Perubahan yang terjadi tidak memiliki pola yang sama. Hal ini dipengaruhi oleh perbedaan warna tiap citra asli. Citra dengan warna yang seragam(intensitas lebih kecil) akan mengalami rasio perubahan yang cenderung lebih kecil, karena semakin beragam warna pada citra, semakin banyak data yang harus ditulis kembali saat proses pembentukan citra stego.

Pengujian Ketahanan Citra

Citra hasil steganografi diuji ketahanannya untuk melihat apakah hasil ekstaksi yang telah disisipkan pesan dapat dilakukan dengan sempurna atau tidak. Pengujian dilakukan setelah citra diberikan derau atau gangguan yang lainnya.

1) Proses Cropping
   Pengujian dilakukan dengan melakukan proses pemotongan citra sebesar 50%. Pesan disisipkan pada citra: “Undira”.

Tabel 6. Tabel Pengujian Cropping

Citra Stego	Citra Stego + Derau	Pesan Terekstrak
		Tidak ada pesan
		Tidak ada pesan
		Tidak ada pesan
		Tidak ada pesan
		Tidak ada pesan

Proses cropping atau pemotongan sebagian citra dapat menghilangkan sebagian pesan yang disisipkan. Pada pengujian ini, citra dipotong sebesar 50%, dimana sebaran pesan bersifat acak dan menyebar pada citra secara merata. Sehingga pemotongan pada citra dapat mengakibatkan pesan yang disisipkan ikut terpotong. Pada pengujian cropping dengan hanya menghilangkan 0.02% dari citra stego, pesan yang disisipkan tetap tidak dapat terekstrak.

Analisa Hasil Pengujian

Dari hasil pengujian diatas maka dapat disimpulkan dan dianalisa :

1) User dapat masuk ke halaman aplikasi apabila mengklik ikon aplikasi yang terdapat di *mobile phone* berbasis Android.
2) Aplikasi ini dapat membuat sebuah pesan rahasia yang hanya bisa diakses oleh pengirim dan penerima saja, untuk membuat sebuah pesan rahasia perlu empat masukan yaitu teks, kunci privat, gambar, dan nama *file*.
3) Fungsi menyimpan berjalan dengan sangat baik apabila ke empat paramaeter dimasukkan.
4) Selain menyimpan pesan rahasia, aplikasi ini juga dapat mengirimkannya dengan cara mengirimkan ke beberapa aksi pengiriman seperti email, *whatsapp, line*, dan lain – lain.
5) Seluruh fungsi dan tombol berfungsi dengan baik sesuai dengan rancangan aplikasi. Aplikasi berjalan dengan baik pada sistem operasi android 2.3.6 (*Gingger Bread*) dan sistem operasi android 4.0.3 (*Ice Cream Sandwidch*). Tidak ada perbedaan tampilan dari kedua sistem operasi android yang diuji.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan Aplikasi Sistem Informasi Terpadu Pelaporan Badan Usaha BBM Non PSO Secara Online (SIPT BBM Non PSO) yang telah dibuat, maka dapat diambil keseimpulan sebagai berikut :

1) Aplikasi SIPT BBM Non PSO yang dibangun merupakan proses pelaporan kegiatan usaha yang dilakukan oleh badan usaha kepada Bandan Pusat.
2) Pada point Ruang Lingkup ada point yang tidak dilaksanakan yaitu, tidak melakukan integrasi sistem NRU dan NRPB karena tidak adanya eksisiting sistem yang berjalan.
3) Badan Pusat dapat melihat seluruh laporan kegiatan usaha yang telah dilaporkan oleh seluruh badan usaha.

Saran

Dalam rangka menjamin keberhasilan dan kelangsungan Sistem Informasi Terpadu Pelaporan Badan Usaha BBM Non PSO Secara Online, pihak Bandan Pusat perlu mengagendakan program kerja, yaitu:

1) Aplikasi NRU dan NRPB agar dikembangkan terlebih dahulu sebelum di integrasikan dengan aplikasi SIPT BBM Non PSO.
2) Program peningkatan kualitas sumber daya manusia berupa pelatihan-pelatihan yang disertai dengan kurikulum serta silabus yang jelas dan tersusun secara terencana.
3) Potensi pengembangan lebih lanjut untuk mengakomodasi pelaporan badan usaha BBM PSO dan badan usaha terkait gas bumi menjadi satu dalam aplikasi SIPT.
4) Penggunaan aplikasi SIPT lebih luas terutama untuk kepentingan pengawasan dan sekretariat.

DAFTAR RUJUKAN

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Systems Analysis and Design: An Object Oriented Approach with UML, 5th Edition. New York: John Wiley & Sons Inc.

Hutahaean, J. (2014). Konsep Sistem Informasi. Yogyakarta: Deepublish.

Irwansyah, E. (2014). Pengantar Teknologi Informasi. Yogyakarta: Deepublish.

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2012). Introduction To Systems Analysis And Design An Agile, Iterative Approach. Joe Sabatino

S. Pressman, Ph. D, Roger diterjemahkan oleh L.N. Harnaningrum. 2002. Rekayasa Perangkat Lunak. Bandung: ANDI.

Permana, Hadi. 2015. Sistem Operasi Terpadu: Optimalisasi Sistem Informasi Dalam Mendukung Kegiatan Usaha Hulu Migas. Konferensi Nasional Sistem & Informatika, STMIK STIKOM Bali.