PENGAMAN DATA UNTUK *SMARTPHONE* BERBASIS *ANDROID* MENGGUNAKAN STEGANOGRAFI ALGORITMA F5

Gunawan Putrodjojo [1]
Markus Lesmana [2]
Dedeh Supriyati [3]

Dosen tetap (DTYM) STMIK Raharja-Tangerang [1)]
Praktisi IT – head PMO pada PT Sinarmas Multifinance [2)]
Dosen tetap (DTYM) STMIK Raharja-Tangerang[3)]

ABSTRAK

Hampir semua bidang kehidupan dan mayoritas kalangan memanfaatkan teknologi dalam pencarian informasi. Adapun teknologi yang sangat banyak digunakan adalah *smart phone* dimana secara fakta hampir semua masyarakat memiliki *smart phone*. Dengan menggunakan perangkat mobile pertukaran segala informasi, berkas, dan pesan bisa berlangsung lebih cepat. Photo merupakan suatu karya pribadi yang biasanya tersebar di dunia maya. Namun, sebenarnya pertumbuhan pemakaian perangkat mobile ini belum diimbangi adanya sistem keamanan data ataupun informasi pada saat pertukaran data atau informasi atau penyimpanannya. Keamanan data dan informasi merupakan suatu hal yang sangat penting apalagi bila data atau informasi tersebut bersifat rahasia. Salah satu cara yang dapat dilakukan untuk mengamankan data atau informasi yang akan dikirim adalah dengan menyembunyikan data atau informasi tersebut ke dalam sebuah wadah (media) agar data atau informasi sulit atau bahkan tidak bisa dikenali oleh indra manusia, atau hal ini biasa digunakan istilah *steganography*(steganografi). Penelitian ini melakukan studi dan implementasi steganografi yang menerapkan Algoritma F5 untuk perangkat mobile yang berbasis Android dimana bahasa pemograman yang digunakan adalah Java dan tool Android Studio dan sebagai pemodelan menggunakan UML(*Unified Modeling Language*). Tujuan penelitiannya adalah mengamankan data dimana pesan yang berupa teks akan dapat disisipkan ke berkas yang berupa JPG atau PNG. Kerahasian pesan-pesan yang dikirim pada arsip berupa gambar yang fungsinya sebagai media perantara sehingga seolah-olah tampak sebagai pesan yang biasa karena pesan yang dikirim hanya bisa dan boleh dibaca penerima yang berhak saja menggunakan kata kunci tertentu.

Kata kunci : Steganografi, Algoritma F5, Android, Bahasa Pemograman Java, , dan UML

ABSTRACT

In almost all fields of life and all circles utilizing technology in searching data and information. The technology is the most used widely are smart phones in which as the fact almost all people have smart phones today. With mobile devices, the exchange of data and information, files, and messages will be able done faster. Photos as personal works wide spreaded in cyberspace. Unfortunately, the growth of mobile devices is not as fast as security system data, information, or exchange and storage growth. The Data and Information security is the most important thing when the information is confidential. One way that we can implement to secure the data or information that we will send is the concealment of data or information into a media so that the data or information more difficult to be recognized by the human senses, or commonly referred to

as steganography. This study or research will implement F5 algorithm in steganography and use Android-based or Android Studio with mobile devices using the programming language Java and the UML(Unified Modeling Language) as modeling method/tool. The purpose of this research was to make the data or information more secure with using steganographic techniques in which a text message will be inserted in PNG or JPG file. Confidentiality of messages sent in an image file that serves as an intermediary medium so that it looked like the usual message because the message is sent only be read by the recipient is entitled to use keywords.
***Key words** : Steganography, Algoritthm F5, Android Operating System, Java, Android Studio, UML Modelling.*

PENDAHULUAN

Pada saat ini *Smartphone* berbasis *Android* bukan lah suatu kemewahan dalam arti sebagai sesuatu yang hanya dapat dimiliki/dinikmati oleh kalangan tertentu. Demgam semakin banyaknya pengguna maka menjadi kan hal yang umum bila melihat anak-anak SD atau tukang becat (hampir semua lapisan masyarakat) menggunakan *Smartphone* dalam kesehariannya.

Tidak jarang pemakai *Smartphone* tidak berhati-hati menggunakan perangkatnya, sehingga sering terjadi perangkatnya jatuh ke tangan orang lain yang tidak bertanggung jawab dan orang lain tersebut bisa dengan leluasa melihat data dan informasi yang terdapat di dalam perangkat *Smart Phone* tadi. Untuk bisa mencegah supaya data dan informasi tidak bisa dimengerti pihak lain diperlukan pengamanan terhadap data dan informasi yang terdapat di dalam *Smart Phone* tersebut.

David Khan dalam bukunya yang berjudul *The Codebreaker : The Story of Secret Writing*, mengatakan "pengamanan data atau informasi dibedakan dalam 2 kelompok yaitu *security* dan *intelligence*. *Security disini* dikaitkan dengan pengamanan data (penting bagi perusahaan) sedangkan kata *Intelligence* dapat dikaitkan dengan pencarian (berupa penyadapan atau pencurian) data (penting bagi militer atau intel/spionase) (Khan D, 1973). Security dapat dilakukan dengan dua cara yaitu *Cryptography* dan *Steganography*(Raharjo B, 2005). Pada prinsipnya, dua-duanya memiliki fungsi yang sama, yaitu berperan dalam keamanan suatu data atau informasi, tetapi memiliki maksud berbeda. *Cryptography* adalah pesan yang dikodekan/ disandikan sedemikian rupa sehingga orang lain tidak dapat mengerti atau tidak dapat mengenali pesan tersebut (Renaldi M, 2006), sedangkan *Steganography* dapat membuat seolah-oleh pesan rahasia tidak ada atau tidak nampak, padahal pesan tersebut ada. Hanya saja kita tidak sadar kalau pesan tersebut ada di sana (Raharjo B, 2005).

Pokok Permasalahan dan Ruang Lingkup Masalah

Untuk mengatasi permasalahan tersebut, perlu kiranya dibuat suatu aplikasi Steganografi dengan Algoritma F5 yang berfungsi menyembunyikan data atau informasi yang berupa pesan ke dalam suatu berkas citra agar pesan tidak bisa dimengerti oleh pihak lain yang tidak berkepentingan dan agar tidak menimbulkan kecurigaan pihak lain. Algoritma F5 dipilih karena karena berdasarkan penelitian-peneltian sebelumnya, algoritma F5 berhasil digunakan untuk meniysipkan pesan ke dalam media gambar menggunakan tipe format JPG tanpa mengubah isi pesan.

Dalam pembuatan aplikasi ini, ada beberapa batasan dan permasalahan

yang ditemukan, diantaranya Aplikasi dibuat dalam bahasa pemrograman Java 1.8, user dapat membuat pesan yang disembunyikan pada berkas citra (steganogram) dan memberikan *password* untuk mengekstraksi, pengguna dapat mengekstraksi berkas citra yang memuat pesan (steganogram) dengan identifikasi berupa *password*, aplikasi yang dibangun dapat dijalankan di *Smart Phone* berbasis Android versi 2.2 (froyo) ke atas, dan Format berkas citra yang bisa digunakan adalah .JPG dan .PNG

Tujuan Fungsional dan Manfaat Penelitian

Adapun tujuan Fungsional penelitian ini adalah agar dapat bermanfaat dan dapat berguna bagi para pengguna *Smart Phone* yang berbasis android untuk meningkatkan keamanan atas data dan/atau informasi yang tersimpan atau terdapat dalam *device* tersebut.

Adapun manfaat penelitian ini antara lain memperluas memperluas wawasan dalam menerapkan teori. Analisis yang dilakukan bisa membantu mengetahui bagaimana sistem aplikasi ini bekerja.

Para pengguna *Smart Phone* bisa memanfaatkan aplikasi ini untuk menambah keamanan pada pesan-pesan yang dikirim maupun yang tersimpan dalam perangkat *Smart Phone* dengan cara menyembunyikan pesan di dalam berkas citra

Pengembangan Aplikasi

Model yang digunakan dalam pengembangan aplikasi ini adalah model OOAD (*Object Oriented Analysis and Design)* yang merupakan pendekatan berbasis objek. OOAD merupakan metode pengembangan sistem yang lebih menekankan objek dibandingkan dengan data atau proses. Ada beberapa ciri utama pendekatan ini, yaitu *object, inheritance* dan *object class.*(Al Fatta H, 2007:31).

Objek merupakan struktur yang meng-enkapsulasi atribut dan metode yang beroperasi berdasarkan atribut-atribut tadi. Objek adalah abstraksi benda-benda nyata dimana data dan proses diletakan bersama agar dapat memodelkan struktur dan perilaku objek-objek dunia nyata.

Objek kelas (*class object*) merupakan sekumpulan objek yang berbagi struktur yang sama dan perilaku yang sama. *Inheritance* merupakan properti yang muncul ketika tipe entitas atau objek kelas disusun secara hirarki dan setiap tipe entitas atau objek kelas menerima atau mewarisi atribut dan metode dari pendahulunya. Untuk memvisualisasi, menspesifikasikan membangun dan pendokumentasian dari pendekatan berbasis objek ini akan digunakan metode UML (*Unified Modeling Language*).

LANDASAN TEORI

Pengertian Steganografi

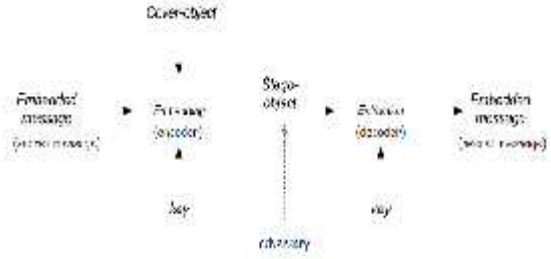
Steganografi (*Steganography*) merupakan ilmu dan seni menyembunyikan pesan rahasia (*hidding message)* sedemikian rupa sehingga keberadaan (eksistensi) pesan tidak dapat terdeteksi oleh indera manusia, padahal pesan tersebut ada. Hanya saja kita tidak sadar kalau ada pesan tersebut di sana (Raharjo B, 2005). Kata steganografi berasal dari bahasa Yunani yang bearti tulisan tersembunyi (*covered writing*). Steganografi membutuhkan 2 (dua) properti yaitu wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung misalnya citra, suara, text atau video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, text atau video.

Steganografi sebenarnya merupakan lanjutan dari Kriptografi dimana dalam kriptografi data yang telah disandikan (*chipertext*) tetap tersedia dan dengan steganografi *chiphertext* tersebut dapat disembunyikan agar pihak ke-3 tidak dapat mengetahui keberadaannya. Di negara-negara yang melakukan penyensoran informasi, steganografi sering dipakai untuk pesan-pesan melalui gambar (*images)*, video atau suara (*audio*).

Penyembunyian data rahasia ke dalam citra digital akan mengubah kualitas citra tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data adalah, (Munir, 2005) :

Fidelity, *Robustness*, dan *Reccovery*. Adapun terminologi yang digunakan dalam proses steganografi adalah *Embedded message* (*hidde text*) atau *secret message*, *Cover-object* (*covertext*), *Stego-object* (*stegotext*), dan *Stego-key*. Teknik-teknik dasar dalam Steganografi *Substitution techniques, Transform domain techniques, Spread spectrum techniques, Statistical techniques, Distortion techniques,* dan *Cover generation techniques,* Contoh metode modifikasi LSB, MSB, dan F5

Secara garis besar, proses steganografi adalah sebagai berikut :

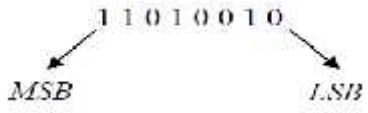


Gambar 1. Diagram Proses Steganografi (Munir, 2005)

Metode-metode (Algoritma-algoritma) Steganografi

Metode/Algoritma *Least Significant Bit* (LSB)

LSB (*Lease Significant Bit*) adalah bit yang paling sedikit pengaruhnya dalam satu *byte.* Selain LSB terdapat MSB (*Most Significant Bit*) yang merupakan bit yang paling besar pengaruhnya dalam satu *byte.*



Gambar 2. Posisi MSB dan LSB pada sebuah *byte*

LSB Merupakan sebuah metode yang lazim digunakan oleh para peneliti pada sebuah steganografi. Karena merupakan metode steganografi yang paling sederhana, cepat, dan mempunyai kapasitas penyisipan suatu informasi digital yang menyisipkan sebuah informasi rahasia pada bit rendah atau *bit* yang paling kanan dari sebuah data *pixel* yang menyusun sebuah informasi digital yang menjadi media penampung suatu informasi rahasia (A. Yogie, P. Andhika dan N. Alfian, 2010).

Metode*Masking* dan *Filtering*

Metode ini biasanya dibatasi pada image 24 bit warna dan image grayscale. Beberapa literatur menyatakan bahwa metode ini mirip dengan watermark, dimana suatu image diberi tanda (*marking*) agar dapat menyembunyikan pesan rahasia. Hal ini dapat dilakukan dengan memodifikasi *luminance image* di beberapa bagiannya. Metode ini memiliki ketahanan

(*robustness*) terhadap kompresi dan *cropping*. Namun, memiliki batasan kapasitas pada informasi yang akan disembunyikan.

Metode *Speed Spectrum*

Metode ini dalam steganografi diilhami dari skema komunikasi *spread spectrum* pada bidang lain untuk mentransmisikan sebuah sinyal pita sempit ke dalam sebuah kanal pita lebar menggunakan penyebaran frekuensi. *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 kompresi dan proses *image* (gambar).

Algoritma F5

F5 merupakan sebuah metode/algoritma yang diajukan oleh Andreas Westfeld dari Techniche Universitas Desden, Institute for System Architecture, Jerman. F5 merupakan perbaikan dari algoritma F3 dan F4.

Algoritma ini tidak menggunakan LSB, namun ia menghitung penyebaran *byte-byte* dari steganogram citra (dalam hal ini berformat JPEG) baik positif maupun negatif, namun bukan 0. Menyisipkan (*embeddin)* bit dari pesan rahasia ke beberapa byte tersebut dengan kompresi XOR(*Exlusive Or*), lalu mengurangi nilai (*decrement*) *byte* tersebut, baik yang disisipi oleh bit dari pesan rahasia maupun tidak.

Kelebihan algoritma ini adalah penyebaran pesannya lebih merata ke seluruh media citra penampung (*cover-image*) karena menggunakan permutasi agar keberadaan pesan sulit untuk terdeteksi (Suhartono, Derwin, dkk, 2012), selain itu F5 menawarkan kapasitas penyimpanan data besar dengan proporsi pesan yang ditampung sebesar 13% dari citra penampungnya (Zulfikar, Dian Hafidh, 2010). Algoritma F5 dapat mencegah serangan statistik dan meningkatkan efisiensi penyisipan karena memiliki 2 fitur utama yaitu *Permutative Straddling* dan *Matrix Encoding* (Kulkarni, Medha, 2012).

Sistem Operasi Android dan Android Studio

Android (/ˈ æn.dr□□d/; an-droyd) adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Sejarah awal Android adalah sebuah perusahaan *software* kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California USA. Didirikan oleh Andy Rubin, Rich Milner, Nick Sears dan Chris White. Menurut Rubin, android didirikan untuk mewujudkan *mobile device* yang lebih peka lokasi dan preferensi pemilik. Dengan kata lain, ingin mewujudkan *mobile device* yang lebih mengerti pemiliknya. Konsep ini ternyata menggugah Google untuk memilikinya, sehingga pada Agustus 2005, akhirnya Google mengakuisisi Android (Supardi Y, 2012).

Android merupakan sistem operasi *open source* dan Google merilis kodenya di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat selular (Supardi Y, 2012). Hal ini memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator seluler dan pengembang aplikasi.

Android Studio merupakan sebuah IDE (*Integrated Development Environment*) yang bisa digunakan untuk pengembangan aplikasi Android dan dikembangkan oleh Google. Android Studio merupakan pengembangan dari Eclipse IDE dan dibuat berdasarkan IDE Java populer yaitu IntelliJ IDEA.

Android Studio direncanakan untuk menggantikan Eclipse kedepannya sebagai IDE resmi untuk pengembangan aplikasi Android. (*Android Studio Overview)*

Konsep Pemodelan *Unfield Modelling Language* (UML)

Pemodelan meupakan gambaran dari realita yang simpel yang dituangkan dalam bentuk pemetaan dengan aturan tertentu (Rosa AS, Shalahuddin, 2015). Pemodelan dapat menggunakan bentuk yang sama dengan realita misalnya bila seorang arsitek ingin memodelkan sebuah gedung yang akan dibangun maka dia akan memodelkannya dengan membuat sebuah maket (tiruan) arsitektur gedung yang akan dibangun dinama maket tersebut dibuat semirip mungkin dengan *design* gedung yang akan dibangun agar arsitektur gedung akan akan dibuat dapat terlihat.

Salah satu perangkat pemodelan adalah *Unifield Modeling Languange* (UML) yang merupakan sebuah bahasa pemodelan yang telah menjadi standar dalam industri *software* untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak (Henderi,2007). Bahasa Pemodelan UML lebih cocok untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi objek (C++, Java, VB.NET). Namun demikian masih bisa tetap digunakan pada bahasa pemrograman prosedural.

UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem, ada beberapa diagram yang disediakan dalam UML (Rosa AS, Shalahuddin, 2015) : Diagram *Use Case*, Diagram Aktivitas, Diagram Sekuensial, Diagram Kolaborasi, Diagram Kelas, Diagram Statechart, dan Diagram komponen,

Metode Pengujian Aplikasi

Ada dua macam pendekatan kasus uji yaitu *white-box* dan *black-box*. Pendekatan *white-box* adalah pengujian untuk memperlihatkan cara kerja produk secara rinci sesuai dengan spesifikasinya (Pressman, 2010). Jalur logika perangkat lunak akan dites dengan menyediakan kasus uji yang akan mengerjakan kumpulan kondisi dan pengulangan secara spesifik. Sehingga melalui penggunaan metode ini akan dapat memperoleh kasus uji yang menjamin bahwa semua jalur independen pada suatu model telah diigunakan minimal satu kali, penggunaan keputusan logis pada sisi benar dan salah, pengeksekusian semua *loop* dalam batasan dan batas operasional perekayasa, serta penggunaan struktur data internal guna menjamin validitasnya. Secara sekilas dapat diambil kesimpulan pendekatan pengujian *white-box* mengarah untuk mendapatkan program yang benar secara 100%.

Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Pressman, 2010). Uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya.

Literature Review

Penelitian tentang steganografi telah dilakukan oleh banyak peneliti. Berikut beberapa penelitian mengenai steganografi dan kesimpulan penelitian-penelitian tersebut.

Tabel 1 Daftar Penelitian Terkait Steganografi

Peneliti	Judul Penelitian	Hasil Penelitian
Viki Miyagina Amal, Alfa Ryano	Aplikasi Steganografi pada Citra Digital	Aplikasi steganografi mampu menyisipkan pesan teks ke dalam

Yohannis pada tahun 2014 dari Institut Teknologi dan Bisnis Kalbis Jakarta	Menggunakan Algoritma Discrete Cosine Transform.	citra digital tanpa menimbulkan perbedaan yang signifikan pada citra tersebut. Jumlah karakter yang dapat ditampung berbeda-beda, hal ini dipengaruhi oleh faktor ukuran gambar dan resolusi gambar.
Maria Magdalena, Nikolaus Adi Putra, Eka Puji Widiyanto dan Willy pada tahun 2015 dari STMIK GI MDP Palembang.	Implementasi Algoritma F5 untuk Penyisipan Pesan Rahasia pada Citra Digital	a)Algoritma dapat digunakan untuk menyisipkan pesan rahasia ke dalam citra digital berwarna (RGB) dengan format JPG dan dapat diterapkan untuk citra resolusi 640x480 sampai dengan 1280 (*width/height*) piksel. Namun, untuk citra dengan resolusi di atas 1280 akan mengalami
		perubahan ukuran (*scalling*) terlebih dahulu agar dapat memenuhi syarat menjadi sebuah *cover image*. b)Citra yang dihasilkan (*stego image*) secara kasat mata tidak jauh berbeda antar *cover image* dan *stego image* dengan angka persentase rata-rata selisih histogram *cover image* dengan *stego image* sebesar 0,23% c)Waktu *encode* pesan bergantung dengan resolusi *cover-image.* Semakin besar resolusi *cover-image*, maka dibutuhkan waktu *encode* pesan lebih lama. d)Stego image tidak tahan (tidak *robust*) terhadap berbagai manipulasi citra, seperti pemberian efek *Gaussian Blur*, merotasi calaitra, merubah ukuran (*scalling*) dan memotong citra (*cropping*). *Stego image* akan gagal ketika di-*decode* untuk mengambil pesan
Ricardo Pramana Suranta pada tahun 2012 dari Sekolah Teknik Elektro dan Informatika Institut Teknologi	Perbandingan Ketahanan Algoritma LSB dan F5 dalam Steganografi Citra.	Steganogram hasil penyisipan dengan algoritma F5 tidak berbeda jauh dengan algoritma LSB. Steganogram dari kedua algoritma tersebut tidak dapat melewati salah satu pengujian ketahanan

Bandung		*watermarking*, yaitu perubahan *contrast*.
I Nyoman Piarsa pada tahun 2010, Staf Pengajar Teknologi Informasi, Fakultas Teknik, Universitas Udaya.	Steganografi Pada Citra JPEG Dengan Metode Sequential dan Spreading	Metode steganografi dengan menggunakan data citra JPEG sebagai media stego merupakan alternatif yang cukup bagus dalam teknik penyembunyian data. Hal ini didukung dengan hasil data citra yang dihasilkan dari proses *embedding* tersebut memiliki tingkat kesamaan yang cukup tinggi dengan citra aslinya, yaitu sebesari 96%. Validitas data ekstraksi yang dimiliki juga mencapai 100%.
Penelitian dari Adhitya Tri Wahyu Utomo pada tahun 2013 dari Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang	Sistem Keamanan Pengiriman Data Pada Email Menggunakan Algoritma F5	a) Aplikasi dapat mengolah citra dengan batas ukuran sebesar 2.02 MB b) Tidak terdapat perubahan terhadap pesan yang disisipkan, hal ini menandakan bahwa pesan yang akan diterima pengguna meruapakn pesan yang benar-benar dikirim oleh pengirim Kapasitas pesan teks yang dapat ditamping dipengaruhi oleh besaran berkas citra yang digunakan, batas maksimal besar teks yang dapat ditampung adalah sebesar 13%dari stego-image yang digunakan

PEMBAHASAN

Kegunaan Steganografi

Seperti perangkat keamanan lainnya, steganografi dapat digunakan untuk berbagai macam alasan, beberapa diantaranya untuk alasan yang baik namun dapat juga untuk alasan yang tidak baik. Untuk tujuan legitimasi dapat digunakan pengamanan seperti citra dengan *watermark* (yang juga dikenal dengan *fingerprinting* yang dikhususkan untuk hal-hal menyangkut *copyright*) sangat mirip dengan steganografi karena menggunakan metode penyembunyian dalam arsip tersebut dan tidak mudah dideteksi oleh kebanyakan orang.

Steganografi juga dapat digunakan sebagai *tag-notes* untuk citra *online*. Terakhir steganografi juga dapat digunakan untuk melakukan penyimpanan atas kerahasian informsi yang berharga, untuk menjaga data tersebut dari kemungkinan sabotase, pencurian atau penggunaan dari pihak yang tidak berwenang.

Hampir semua berkas *digital* dapat digunakan sebagai media steganografi tetapi format yang paling cocok adalah yang mempunyai nilai *bits redundancy* tinggi. *Bits redundancy* adalah bit yang dapat diubah tanpa merubah banyak karakteristik file secara keseluruhan. Berkas citra dan suara adalah yang memenuhi syarat ini, sehingga banyak peneliti steganografi menggunakan media tersebut.

File Citra pada komputer merupakan array bilangan yang merepresentasikan nilai intensitas cahaya yang bervariasi (*pixel*). Kumpulan *pixel* inilah yang membentuk suatu citra. Citra yang sering digunakan umum adalah citra 24 bit dan citra 8 bit (256 colors), (Johnson, 1998).

Teknik steganografi gambar dapat dibagi menjadi dua bagian yaitu *spatial domain* dan *transform*/*frequency domain*. Pada *spatial domain* informasi dimasukkan ke dalam tiap pixel satu per satu. Sementara itu, pada *transform domain*, gambar ditransformasikan terlebih dahulu kemudian informasi baru dimasukkan ke gambar.

Teknik steganografi pada *spatial domain* menggunakan metode*bit-wise* yang menggunakan penyisipan *bit* dan*noise manipulation*.Format gambar yang paling cocok untuk cara ini adalah tipe *lossless*. Namun, cara ini sangat bergantung kepada format gambarnya, (Morkel, dkk, 2005).

Algoritma Yang Digunakan

Kelebihan dari algoritma F5 adalah penyebaran pesannya lebih merata ke seluruh media citra penampung (*cover-image*) karena menggunakan permutasi sehingga keberadaan pesan sulit untuk terdeteksi (Suhartono, Derwin, dkk, 2012), selain itu F5 menawarkan kapasitas penyimpanan data besar dengan proporsi pesan yang ditampung sebesar 13% dari citra penampungnya (Zulfikar, Dian Hafidh, 2010). Algoritma F5 dapat mencegah serangan statistikdan meningkatkan efisiensi penyisipan karena memiliki 2 fitur utama yaitu *Permutative Straddling* dan *Matrix Encoding* (Kulkarni, Medha, 2012).

Proses Penyisipan (*Embedding*)Algoritma F5

Algoritma F5 meng-*embed* bit pesan ke koefisien DCT yang dipilih secara acak dan menggunakan matriksembedding yang meminimalkan jumlah perubahan yang perlu untuk menanamkan panjangpesan tertentu.Proses embedding dimulai dengan menurunkan benih untuk PRNG (Pseudo RandomNumber Generator) dari kata sandi pengguna dan menghasilkan “random walk” koefisienDCT dari cover image tersebut. PRNG juga digunakan untuk mengenkripsi nilai kmenggunakan stream cipher dan menanamkannya dalam cara yang teratur bersama-samadengan panjang pesan di awal aliran pesan. Tubuh pesan tertanam menggunakan embeddingmatriks, menyisipkan k bit pesan ke satu kelompok 2k-1 koefisien dengan menurunkan nilaiabsolut paling banyak satu koefisien dari masing-masing kelompok satu. Proses embeddingterdiri dari langkah-langkah berikut:

1. Ambil nilai RGB dari gambar input
2. Hitung tabel kuantisasi yang sesuai dengan faktor kualitas Q dan kompres gambar saatmenyimpan DCT terkuantisasi koefisien.
3. Hitung perkiraan kapasitas tanpa embedding matriks C = hDCT - hDCT / 64 - h (0) - h (1)+ 0.49h (1), di mana hDCT adalah jumlah semua koefisien DCT, h (0) adalah jumlahkoefisien DCT AC bernilai nol, h (1) adalah jumlah dari AC Koefisien DCT dengannilai absolut 1, hDCT/64

adalah jumlah dari DC koefisien. Parameter C dan panjangpesan yang digunakan untuk menentukan matriks embedding terbaik.

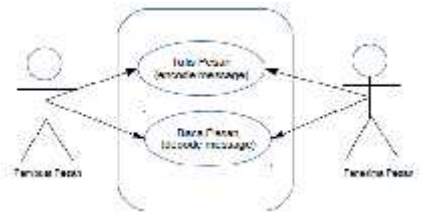
4. Password yang ditentukan pengguna digunakan untuk menghasilkan benih untukPRNG juga digunakan menentukan jalur acak untuk embedding bit-bit pesan. PRNGjuga digunakan untuk menghasilkan pseudo-random bit-stream yang diXOR denganpesan untuk membuatnya bit-stream teracak. Selama embedding, koefisien DC dankoefisien sama dengan nol dilewati.
5. Pesan dibagi menjadi segmen-segmen dari k bit yang tertanam ke dalam kelompok 2k-1koefisien sepanjang jalur acak. Jika hash dari kelompok yang tidak cocok dengan bit-bitpesan, nilai absolut dari salah satu koefisien dalam kelompok diturunkan satu untukmendapatkan nilai yang cocok. Jika koefisien menjadi nol, kejadian ini disebut sebagaipenyusutan, dan k bit pesan yang sama diembed ulang dalam kelompok berikutnya darikoefisien DCT.
6. Jika ukuran pesan sesuai dengan perkiraan kapasitas, maka proses embed berlanjut, lain daripada itu error yang menunjukkan panjang maksimal yang mungkin akan ditampilkan.

Implementasi Sistem

Implementasi sistem akan dijelaskan melalui :Use Case Diagram, Diagram Aktivitas, dan Diagram Sekuen

Use Case Diagram

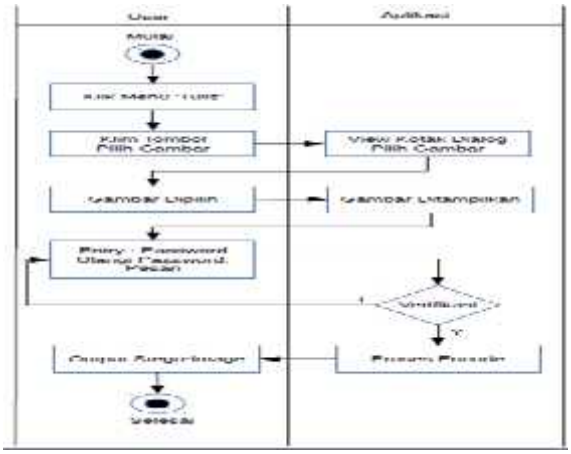
Pada diagram *use case* (gambar 3.), *user* pengirim pesan dan penerima pesan akan memiliki menu-menu yang sama. Kedua-duanya dapat membuat dan mengekstraksi pesan.



Gambar 3. Use Case Diagram

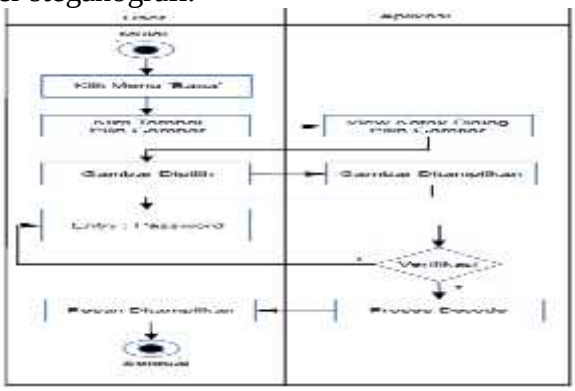
Diagram Aktivitas

Diagram proses aktivitas pada aplikasi ini dibagi menjadi 2 (dua), yaitu : Proses pembuatan berkas *stego-image* (gambar 4.) dan proses ekstraksi berkas *stego-image* (gambar 5.).



Gambar 4. Diagram Aktivitas Pembuatan Berkas *Stegano-Image*

Proses proses pembuatan *stego-image* (*encode message*), dimulai dari pemilihan menu ‘Tulis’ pada aplikasi yang dilanjutkan dengan pemilihan *cover-image* yang sudah tersedia pada perangkat *mobile*. Aplikasi akan menampilkan *cover-image* yang dipilih. Langkah selanjutnya, pengguna memasukan kata sandi (yang akan digunakan untuk mengekstrak pesan yang tersebunyi pada *cover-image*), menginput ulang kata sandi dan memasukan pesan pada tempat yang disediakan. Langkah terakhir adalah menekan tombol proses yang memerintahkan aplikasi untuk melakukan proses *encode*. Berkas *stego-image* hasil dari proses ini akan disimpan pada folder steganografi.

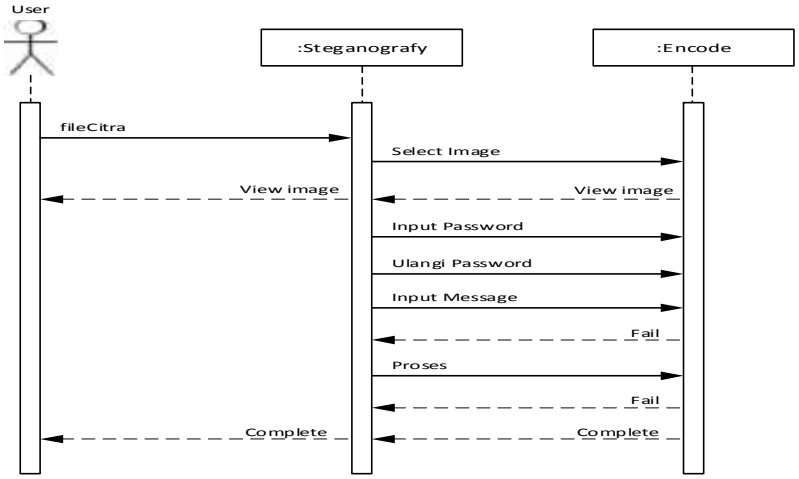


Gambar 5. Diagram Aktivitas Ekstraksi Berkas *Stegano-Image*

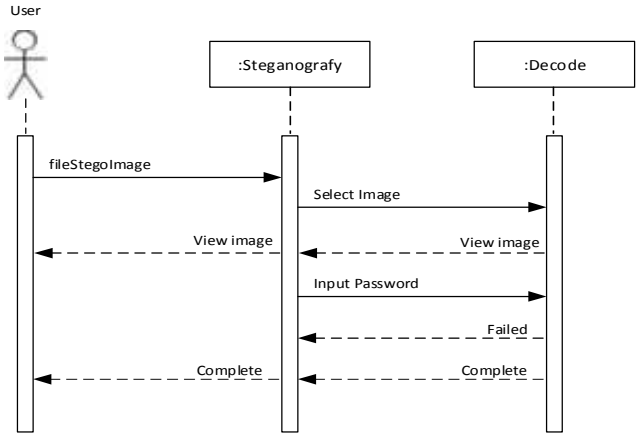
Proses ekstraksi (*decode*), dimulai ketika pengguna memilih menu ‘Baca’ yang dilanjutkan dengan menekan tombol ‘Pilih Gambar’. Sistem akan menampilkan kotak dialog untuk memilih gambar, setelah gambar dipilih sistem akan menampilkan gambar tersebut pada tempat yang disediakan. Selanjutnya pengguna memasukan sandi dan menekan tombol ‘Proses’. Aplikasi akan memproses dan menampilkan pesan jika kata sandi yang dimasukan cocok dengan kata sandi yang tersimpan pada *stego-image* tersebut.

Diagram Sekuen

Proses yang terjadi pada aplikasi ini adalah *Encode* pesan ke dalam berkas citra sehingga menjadi sebuah berkas *stego image* (gambar 6.) dan proses *decode* untuk menampilkan pesan yang ada pada *stego image* (gambar 7.).Berikut adalah diagram sekuen untuk aplikasi ini :



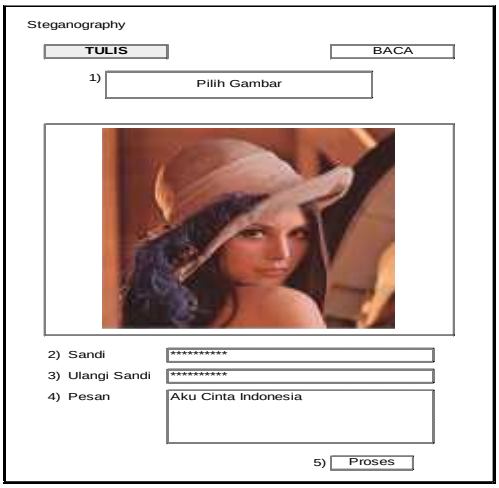
Gambar 6. Diagram Sekuen Pembuatan Berkas *Stego-Image*



Gambar 7. Diagram Sekuen Ekstraksi Berkas *Stego-Imag*

Antarmuka Aplikasi

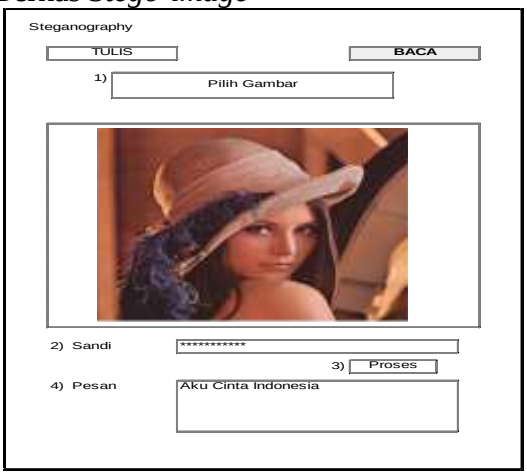
Antarmuka (*interface*), *user interface* yang ditampilkan adalah tombol pilih Tulis dan Baca, Pilih Gambar. Secara *default* pilihan berada pada 'Tulis'. Tampilan Membuat Berkas *Stego-image*



Gambar 8. Tampilan Aplikasi Membuat Berkas *Stego-image*

Langkah-langkah untuk membuat steganogram :Pilih Gambar, setelah gambar dipilih maka aplikasi akan menampilkan gambar tersebut pada bidang yang telah disedikakan. Selanjutnya pengguna memasukan *password*, menginput ulang *password* dan memasukan pesan pada bidang yang telah disediakan. Langkah terakhir adalah menekan tombol proses, maka aplikasi akan melakukan proses *decode* dan membuat berkas *stego-image.*

Tampilan Ekstraksi Berkas *Stego-image*



Gambar 9. Tampilan Aplikasi Ekstraksi Berkas *Stego-image*

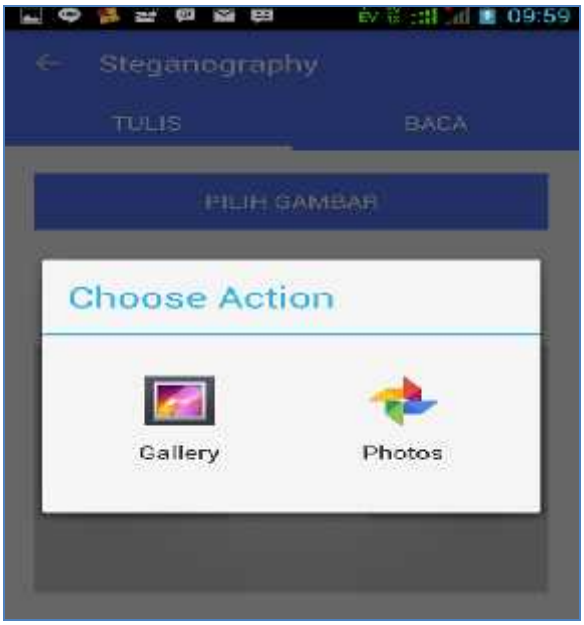
Untuk mengekstraksi pesan, langkah pertama harus memilih menu ‘Baca’, langkah-langkah selanjutnya memilih berkas *stego image,* masukan *password* dan menekan tombol proses.Aplikasi akan memproses dan menampilkan pesan pada bidang yang telah disediakan password benar.

IMPLEMENTASI DAN TESTING

Implementasi

Perangkat Keras yang digunakan untuk mengimplementasikan sistem adalah Smart Phone dengan spesifikasi hardware sebagai berikut : Processor QuadCore 1.2 Ghz, RAM 1 GB, dan HD Resolution 720x1280 pixels. Perangkat Lunak yang digunakan untuk mengimplementasikan sistem adalah sebagai berikut Android 4.2.1 Jelly Bean.

Implementasi Antarmuka terbagi menjadi 2 (dua) bagian, yaitu antarmuka untuk menyembunyikan pesan (*encode*) dan mengekstraksi pesan (*decode*) dalam 1 (satu) aplikasi. Jadi, ketika aplikasi ini dijalankan maka pengguna dapat menyembunyikan pesan atau mengekstraksi pesan.



Gambar 10. Pilih Gambar (*Cover Image*)

Pada proses ini (gambar 10.), pengguna diminta untuk memilih gambar yang akan dijadikan *cover-image* yang dapat diambil dari *foldergallery* atau *photos* yang tersedia diperangkat pengguna. Setelah gambar dipilih maka akan ditampilkan (gambar 4.3)

Gambar 11 *Cover Image* ditampilkan

Pada tahap ini (Gambar 11.) aplikasi menampilkan gambar yang telah dipilih untuk dijadikan *cover-iamge* beserta nama dan lokasi penyimpanan berkas tersebut.

Gambar 12. Menulis *Password* dan Pesan

Langkah selanjutnya adalah memasukan Password (kata sandi) dan pesan yang akan disisipkan pada berkas gambar. Pada tahap ini pengguna diharuskan untuk mengisi kata sandi, mengulangi kata sandi dan menulis pesan. Selanjutnya pengguna mengetuk tombol tulis untuk proses *embeded* pesan. Berkas *stego-*

image disimpan pada *folder* steganografi dengan nama berkas : nama *cover-image* + "_"

Proses ekstraksi pesan dimulai dengan cara pengguna mengetuk tombol baca.

Setelah pengguna mengetuk tombol baca, aplikasi akan menampilkan jendela *Choose Action* untuk memilih gambar yang ada pada folder *gallery* atau *photos.* Setelah berkas gambar dipilih maka aplikasi akan menampilkan gambar dari berkas tersebut beserta nama berkas dan lokasi penyimpanan berkas.

Pengujian

Metode pengujian yang akan digunakan untuk menguji sistem adalah metode pengujian *Black Box.* Metode pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak.

Hasil pengujian terhadap fungsi pembuatan berkas :

Tabel 2. Pengujian Pembuatan *Stego-image*

Nama Task	Fungsi	Pencapaian
Pilih Gambar	Memasukan gambar yang akan dijadikan penampung dari pesan.	Fungsi berjalan dengan baik dimana berkas citra yang ditampilkan dari internal dan eksternal *storage*
Sandi	Memasukan password yang dipakai untuk mengekstraksi pesan.	Kata sandi berhasil dimasukan dengan maksimal 10 karakter
Ulangi Sandi	Memasukan ulang kata sandi yang sama	Pengulangan berjalan sesuai fungsi.
Pesan	Memasukan pesan yang akan di-encode ke dalam berkas citra.	Pesan berhasil ditulis dengan jumlah karakter maksimal 255.
Tombol Tulis	Proses untuk membuat berkas steganogram.	Fungsi dapat digunakan sebagaimana mestinya dan berkas steganogram disimpan pada folder steganography.

Hasil pengujian terhadap fungsi ekstraksi pesan :

Tabel 3. Pengujian Ekstrasi Berkas *Stego-image*

Nama Task	Fungsi	Pencapaian
Pilih Berkas Steganogram	Memasukan berkas steganogram yang akan diekstraksi	Berkas steganogram harus dipindahkan ke folder image default yang dikenal oleh system android
Sandi	Masukan password yang akan diverifikasi untuk mengekstraksi berkas steganogram	Proses tidak berjalan bila password salah
Tombol Baca	Proses ekstraksi pesan	Ekstraksi pesan ditampilkan bila

		password benar

Tabel 4 Pengujian *Encode* Pesan dengan karakter yang sama

	Nama File	Resolusi	Ukuran	Resolusi StegoImage	Ukuran StegoImage	Proses Ekstraksi
1.	Bbm-me.jpg	640x640	61 Kb	640x640	54 Kb	Sukses
2.	Lena.jpg	512x516	101 Kb	512x512	87 Kb	Sukses
3.	Gunung.jpg	1365x1026	367 Kb	1365x1026	258 Kb	Sukses

Gambar 13. Lena.JPG asli

Gambar 14. Lena.JPG disisipi pesan 12 chr

Gambar 15. Gunung.JPG asli

Gambar 16. Gunung.JPG disisipi pesan 12 chr

- Hasil pengujian dengan berkas yang sama dan panjang pesan yang berbeda :

Tabel 5. Pengujian *Encode* Pesan dengan berkas *cover image* yang sama

No.	Nama File	Resolusi	Ukuran	Panjang Pesan	Ukuran StegoImage	Proses Ekstraksi
1.	Bbm-me.jpg	640x640	61 Kb	50 karakter	54 kb	Sukses
2.	Bbm-me.jpg	640x640	61 Kb	255 karakter	54 Kb	Sukses

KESIMPULAN

Algoritma Steganografi F5 berhasil digunakan untuk menyisipkan pesan ke dalam media gambar dengan format tipe data JPG dan PNG tanpa mengubah isi pesan dan berhasil meningkatkan keamanan data dari pihak luar yang tidak berkepentingan. Program aplikasi juga dapat menerapkan Steganografi Algortima F5 dan dapat berjalan dengan baik dalam perangkat mobile berbasis Android. Berkas *stego-image* memiliki ukuran berkas yang lebih kecil dibanding berkas asli.

Dalam proses steganografi dengan metode F5 ini masih terdapat beberapa aspek yang dapat dikembangkan lebih lanjut. Beberapa diantaranya yaitu :

1. Pengembangan lebih lanjut untuk memakai tipe data yang lain selain tipe data JPG dan PNG.

2. Program aplikasi dapat dikembangkan lebih lanjut agar dapat langsung melakukan pengiriman pesan yang telah tersisipi.
3. Berkas Stegao-Image dapat dikirim pengiriman pesan yang tidak mereduksi berkas citra seperti WhatsApp dan sebagai lampiran dalam email.
4. File stego-image tidak tahan terhadap proses mengeditan seperti : *cropping*, *resize* dan *rotate*.
5. Mengembangkan perangkat keras (*hardware*) yang lebih mendukung kecepatan proses steganografi pada perangkat mobile.

DAFTAR PUSTAKA

[1] Android Studio Overview, http://developer.android.com/tools/studio/index.html diakses 19 Februari 2016

[2] Al Fatta, H. 2007, Analisis & Perancangan Sistem Informasi, CV Andi Offset

[3] Alfan, Christian dan Rojali, 2012. Perancangan Program Aplikasi Penyembunyian Pesan pada Citra JPEG dengan Algoritma F5 dalam Perangkat Mobile Berbasis Android, Thesis Binus

[4] Yogie, A., Pratama A., dan Nurlifa A., 2010. Studi Pustaka Untuk Steganografi Dengan Beberapa Metode, Fakultas Teknologi Industri, Universitas Islam Indonesia.

[5] Clune, T.L., and R.B. Rood, 2011, *Software testing and verification in climate model development*, IEEE Journal, Focus: climate change software, September-October, pp. 49-55.

[6] Henderi, 2007.*Analysis and Design System with Unfield Modelling Language (UML)*, STMIK Raharja, Tangerang

[7] Jin, J., and Xue, F., 2011.*Rethinking software testing based on software architecture*, in IEEE Proceeding of 7th International Conference on Semantics, Knowledge and Grids, pp. 148-151. DOI 10.1109/SKG.2011.32

[8] Jubilee Enterprice, 2015, Mengenal dasar-dasar pemrograman Android, Elex Media Komptinto

[9] Kadir, A., 2014, Buku Pertama Belajar Pemrograman Java, Mediakom.

[10] Khan, D., 1973, *The Codebreaker : The Story of Secret Writing*, New American Library.

[11] Kumamoto, H., 2010, *Destructive testing of software systems by model checking*, IEEE Journal, pp. 261-266.

[12] Munir R., 2006, Kriptografi, Informatika

[13] Morkel, T., Eloff, J.H.P., dan Oliver, M.S., 2005, *An Overview of ImageSteganography,* ICSA Research Group, Department Computer Sceince Pretoria, Afrika Selatan.

[14] Pranoto B., 2011. Steganografi Pada Citra Digital Menggunakan MetodeSpread Spectrum dan Metode Least Significant Bit (LSB) Modification,Tugas Akhir Universitas Islam Negeri Sultan Syarif Kasim, Riau.

[15] Pressman, R.S., 2010.*Software Engineering: a practitioner's approach,* 7th Edition, McGraw-Hill,New York.
[16] Rahardjo B., 2005. Keamanan Sistem Informasi berbasis Internet. PT Insan Infonesia – Bandung dan PT INDOCISC – Jakarta. pp. 31
[17] Rosa A.S, Shalahuddin. 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, Informatika.
[18] Santoso, Harip, 2010. Aplikasi Web/asp.net, Elex Media Kompitindo, Jakarta
[19] Supardi Y, 2012.Sistem Operasi Andal Android, PT Elex Media Komputindo.
[20] Suhartono, Derwin, dkk, 2012. Aplikasi Penyembunyian Pesan pada Citra JPEG dengan Algoritma F5 dalam Perangkat Mobile Berbasis Android, Seminar Nasional Aplikasi Teknologi Informasi 2012 (SNATI 2012), Yogyakarta, 15-16 Juni.
[21] Supriyanto, 2005.Perancangan Aplikasi, Widyastana, Surabaya.
[22] Westfeld A, *F5 A Steganograpic Algortihm, High Capacity Despite Better Steganalysis*
[23] Widianti, Sri 2000. Pengantar Basis Data tentang Aplikasi dan DBMS,Fajar, Jakarta.
[24] Xie, T., 2011.*A study on methods of software testing based on the design models*, in Proceeding of 6th International Conference on Computer Science and Education (ICCSE 2011), August 3-5, Singapore, pp. 111-113.
[25] Zulfikar, Dian, H., 2010. Uji Ketahanan Algoritma F5 pada Stego Image Terhadap Image Distortion, Skripsi, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Maulana Malik Ibrahim, Malang.