Pengamanan Pesan Text Menggunakan Metode Steganografi Least Significant Bit dengan Media Digital Gambar

Ragil Mulyono, A11.2011.06104[1], Erna Zuni Astuti[2]

Program Studi Teknik Informatika – S1

Fakultas Ilmu Komputer

Universitas Dian Nuswantoro, Jl. Nakula I No. 5-11, Semarang
111201106104@mhs.dinus.ac.id[1], afis@dosen.dinus.ac.id[2]

Abstrak

Semakin berkembangnya teknologi komunikasi pada erah saat ini kebutuhan akan keamanan dalam berkomunikasi jadi sangat penting maka dikembangkan motode steganography yang tidak hanya dapat menyembunyikan pesan tapi juga dapat mengelabuhi dari para penyadap pesan dengan kemasan kemasan yang tidak mencurigakan.Penelitian ini bertujuan untuk mengamankan data dengan metode steganography dan teknik least significant bit pada image, pesan teks akan disisipkan pada berkas jpeg,png,dan gif. Bit –bit pesan akan di embededing kan ke media dengan merubah nilai bit terakhir, dengan bantuan *pseodo random* bit bit pesan akan tersusun secara acak namun tetap dapat terekstraksi.Metode least Significant Bit bekerja dengan merubah 1 bit terakhir pada bit stego-gambar. Kerahasiaan pesan yang dikirim pada file gambar yang berfungsi sebagai media sehingga tampak seperti pesan biasa melalui aplikasi sosial media yang terinstall pada device android user, karena pesan yang dikirim hanya dapat dibaca oleh pengirim dan penerima yang memiliki aplikasi serupa. Aplikasi yang dibuat dalam penelitian ini menggunakan Software pendukung Java Developmen Kit (JDK), IDE Eclipse (Integegrated Development Environment), Android Software Development Kit (SDK), Android Development Tools (ADT) Plugins. Diimplementasikan pada telepon sellular berbasis android 4.4 dan API Level 19.Sehingga diperoleh software aplikasi yang dapat diinstal pada smartphone yang memiliki komputasi cukup baik.

Abstract

The continued development of communication technology in today's era of security needs to communicate so it is very important it motode steganography developed which can not only hide messages but can also fool of tappers message with packaging that does not mencurigakan.Penelitian packaging is intended to secure the data by the method of steganography and engineering least significant bit of the image, a text message will be inserted in the file jpeg, png, and gif. Bit-bit message will embededing right to media by changing the value of the last bit, with the help of random bits pseodo message bits will be arranged randomly but still be in ekstraksi.Metode least significant bit works by changing one bit last bit stego-image. Confidentiality of messages sent on the image file that serves as the media so that it looks like a normal message through social media applications installed on the Android device users, because the message sent can only be read by the sender and recipient have the same application. Applications are made in this study using Java supporting software developmen Kit (JDK), Eclipse IDE (Integegrated Development Environment), Android Software Development Kit (SDK), the Android Development Tools (ADT) Plugins. Implemented on a cellular phone based on Android 4.4 and API Level 19 to obtain software applications that can be installed on smartphones that have a good enough computing.

1.1 Latarbelakang

Metode Least Significant Bit adalah metode penyisipan (steganografi) kedalam media lain, yang mana metode ini sudah digunakan penelitian penelitian sebelumnya. Penelitian

yang di lakukan oleh Tri cahyadi tahun 2012 dengan judul, " Implementasi steganografi LSB dengan viginere cipher pada citra jpeg", dengan hasil dari gabungan metode kriptografi vigenere dan steganografi LSB menghasilkan hasil yang cukup baik. Penelitian yang dilalkukan oleh Hapsari M pada tahun 2010 dengan judul "Studi steganografi pada image file" dengan hasil LSB pada *gif adalah algoritma yang sangat efektif untuk digunakan ketika mengembeded pesan ke dalam citra grayscale. Penelian selanjutnya oleh Oster Dwi Merbial pada tahun 2012 dengan judul "implementasi steganografi untuk menyembunyikan gambar dalam audio menggunakan LSB", dengan hasil dengan meode lsb pesan mampu menggunakan kapasitas maksimal dalam penyembunyian pesan. Dari beberapa penelitian terkait yang dapat dianalisa bahwa metode LSB ini baik untuk metode penyembunyian pesan kedalam suatu media.

Pada Tugas Akhir ini, akan dirancang dan diimplementasikan steganografi pada gambar digital. Untuk menyembunyikan data pada dokumen gambar, digunakan metode *Least Significant Bit*. LSB merupakan metode untuk menyembunyikan pesan di dalam *file* gambar. Metode ini menggunakan *bit* yang ada di dalam *file* gambar untuk menyembunyikan pesan atau informasi. Informasi atau pesan akan disembunyikan dengan mengacak bit terakhir dari setiap warna. Perubahan bit tersebut diatur sedemikian rupa untuk mengelabuhi penglihatan manusia sehingga tidak mudah untuk dideteksi [4]. Berdasarkan latar belakang masalah yang telah disebutkan diatas maka judul yang akan di usulkan *Pengamanan Pesan Text Menggunakan Metode Steganografi Least Significant Bit dengan media digital gambar*.

1.2 Rumusan Masalah

Bagaimana mengamankan pesan text dengan cara menyisipkan pesan pada media digital gambar dengan menggunakan algoritma Least Significant Bit pada perangkat mobile berbasis android.

1.3 Tujuan Penelitian

1. Mengamankan pesan text melalui media gambar sebagai inangnya .
2. Merancang dan mengimplementasikan perangkat lunak steganografi pada gambar digital yang dapat menyembunyikan data berupa text kedalam gambar digital dengan metode LSB.

Landasan Teori

2.1 Steganografi

Steganografi berasal dari bahasa Yunani yaitu *Steganos* yang berarti menyembunyikan dan *Graptos* yang bereti tulisan sehingga steganografi tulisan yang disembunyikan [3]. Secara umum steganografi adalah teknik menyisipkan pesan kedalam suatu media [7]. Walaupun steganografi dapat dikatakan mempunyai hubungan erat dengan kriptografi, tetapi metode ini sangat berbeda. Kriptografi mengacak pesan sehingga tidak dimengerti, sedangkan steganografi menyembunbyikan pesan sehingga pesan tidak terlihat. Pesan yang diacak dengan metode kriptografi mungkin akan menimbulkan kecurigaan, namun untuk tidak untuk pesan yang dibuat dengan steganografi.

2.2 Least Significant Bit

Secara umum, terdapat dua proses didalam steganografi. Yaitu proses embeding untuk menyembunyikan pesan dan ekstrasi untuk meng ekstrasi pesan yang disembunyikan. Proses-proses tersebut dapat dilihat pada gambar berikut ini:

Sebagai ilustrsi misalkan *cover-object* adalah citra sekempulan citra berwarna merah seperti yang terlihat pada contoh di bawah ini:

00110011 10100010 11100010 01101111

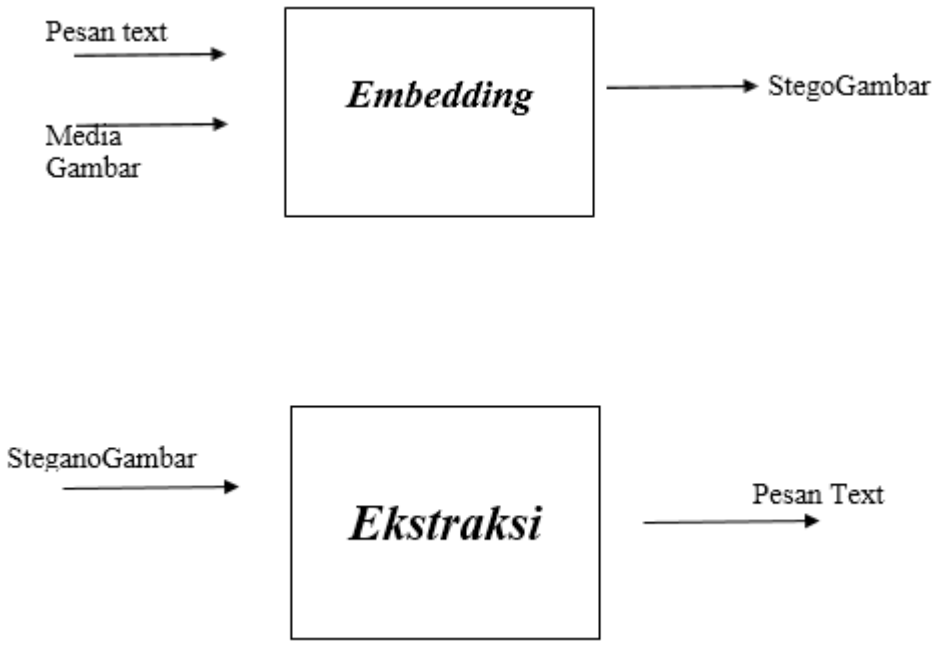
Dan misalkan pesan rahasia (yang telah dikonversi ke system biner) *embedded message* adalah 0110. Setiap bit dari *watermark* menggantikan posisi LSB dari segmen *pixel-pixel* citra menjadi :

00110010 10100011 11100011 01101110

Dari hasil penanaman atau *embedding* kedalam sekumpulan *pixel* citra berwarna merah tadi diperoleh kembali sekumpulan *pixel* berwarna merah yang telah berubah sedikit pada posisi bit terendah atau LSB dari pixel tersebut. Demikianlah contoh sederhana bagaimana algoritma LSB bekerja untuk menggantikan nilai bit-bit terendah dari setiap *pixel* untuk disisipkan atau digantikan oleh bit baru yang mengandung pesan.

Untuk dapat membuat *hiddentext* tidak dapat dilacak, bit-bit pesan tidak mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit *LSB-nya* dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49. Pembangkitan bilangan acak dilakukan dengan *pseudo-random-number-generator* (*PRNG*) yang berlaku sebagai kunci stegano.

Pada citra 8-bit yang berukuran 256 x 256 *pixel* terdapat 65536 *pixe*l, setiap *pixel* berukuran 1 *byte* sehingga kita hanya dapat menyisipkan 1 bit pada setiap *pixel*. Pada citra 24-bit yang berukuran 256 x 256 *pixel*, satu *pixel* berukuran 3 *byte* (atau 1 *byte* untuk setiap komponen R, G, B), sehingga kita bisa menyisipkan pesan sebanyak 65536 x 3 bit = 196608 bit atau 196608/8 = 24576 *byte*. Pesan yang disembunyikan di dalam citra dapat diungkap kembali dengan mengekstraksinya. Posisi *byte* yang menyimpan bit pesan dapat diketahui dari bilangan acak yang dibangkitkan oleh *PRNG*. Jika kunci yang digunakan pada waktu ekstraksi sama dengan kunci pada waktu penyisipan, maka bilangan acak yang dibangkitkan juga sama. Dengan demikian, bit-bit data rahasia yang bertaburan di dalam citra dapat dikumpulkan kembali. Berikut ini ilustrasi proses yang terjadi steganografi secara umum:



Gambar 2.2 Embeding & Ekstraksi Pesan

Keterangan :

: input/output

Gambar 2.2 Embeding menunjukkan proses penyembunyian pesan dimana dibagian pertama, dilakukan proses embedding pesan text yang hendak disembunyikan secara rahasia kedalam media gambar sebagai penyimpanan, sehingga dihasillkan media dengan data tersembunyi didalamnya (StegoGambar).

Gambar 2.2 Ekstraksi dilakukan proses ekstrasi pada StegoGambar. Kemudian kebanyakan teknik steganografi, ekstrasi pesan tidak akan mengembalikan media (gambar) awal persis sama dengan media setelah dilakukan ekstrasi bahkan sebagian besar mengalami kehilangan. Karena saat penyimpanan pesan tidak dilakukan pencatatan kondisi awal dari media yang digunakan untuk penyimpanan pesan.

Contoh Kasus:

Persamaan Utama dari LSB ini adalah

$$S(i,j) = C(i,j) - 1, \text{ if } LSB(C(i,j)) = 1 \text{ and } m = 0$$

$$S(i.j) = C(i,j), \text{ if } LSB(C(i,j)) = m$$

$$S(i,j) = C(i,j) + 1, \text{ if } LSB(C(i,j)) = 0 \text{ and } m = 1$$

Dimana, LSBC(i,j) adalah cover-image dan C(i,j) adalah pesan text yang disembunyikan. S(i,j) adalah stego-image.

Contoh kasus, untuk menyembunyikan pesan ke dalah gambar dengan ukuran 3 pixel (24-bit RGB). Bit dari 3 pixel ini adalah :

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

Progam dapat menyembunyikan huruf "J" yang mana posisinya berada no 74 dalam tabel ASCII ya itu "01001010", dan akan disisipkan menjadi :

(11101010 11101001 11001010)

(01100110 11001011 11101000)

(11001001 00100100 11101001)

Bit berubah setelah di embediing didalam stego-image, selanjutnya gambar siap dikirim.

III Metode Penelitian

3.1 Metode Penyisipan

Proses penyisipan menggunakan Metode LSB mempunyai langkah langkah sebagai berikut, pertama adalah sistem meng-load file gambar RGB (media gambar digital) dari smartphone, kemudian gambar diekstrak per pixsel sampai menjadi satuan terkecil(bit) misal 11100010, selanjutnya menghitung kapasitas gambar untuk media, besarnya kapasitas media untuk cover stego adalah 3 kali jumlah pixsel dari gambar stego. Untuk mengacak bit yang berisi pesan pada cover image stegano dibutuhkan pengacak yang terstruktur atau generator kunci stegano yang disebut *Pseudo-random*, yang disini di implementasikan *pseduo-number-squence* salah satu dari metode *pseuodo random.* Selanjutnya akan dibaca semua bit yang ada di gambar, selagi dibaca di siapkan text yang akan sisispkan pada media gambar. Setelah semuanya siap, pesan text akan sisipkan ke media gamabar dengan metode LSB. Kemudian akan di transformasikan kembali

menjadi gambar RGB. Setelah gambar stego-image terbentuk gambar siap ditampilakan maupun di distribusikan.

3.2 Metode Ekstraksi

proses ekstraksi menggunakan Metode LSB dan Deskripsi pesan menggunakan Algoritma LSB yang mempunyai langkah langkah sebagai berikut, pertama inputan berupa gambar yang di*support* atau di dukung oleh perangkat lunak. Gambar akan dibaca dan dirubah menjadi susunan terkecil(bit) misal 11100010, kemudian di cek kunci dengan metode pseudo random, apakah ada kuncinya? Jika tidak maka akan gagal yakni tidak ada pesan yang di tampilkan, jika ada maka akan dibangkitkan pseudo random untuk membaca seluruh bit-bit gambar sehingga didapatkan bit pesan yang berisi karakter. Bit bit pesan di baca sesuai urutan dari pseudo random kemudian bit bit pesan di ekstraksi menjadi karakter dan di tampilkan.

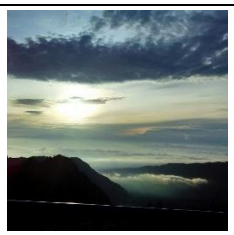
3.2 Eksperimen dan Cara uji

Eksperimen dilakukan dengan pembuatan program android dengan menggunakan ECLIPSE Android Developer Tools. Setelah Stego-Gambar diperoleh, kemudian kualitas Stego-Gambar diukur menggunakan PNSR. Dari nilai PNSR ini akan diketahui apakah Gambar yang berisi informasi pesan tersebut *robust* atau tidak.

Dari proses penyisipan pesan ke dalam file citra tentunya akan ada perbedaan kualitas citra sebelum dan sesudah proses penyisipan pesan, untuk mengetahui seberapa besar penurunan kualitas citra maka akan dilakukan perhitungan nilai PSNR seperti yang telah dijelaskan pada bab sebelumnya.

Citra pengujian memiliki ukuran yang bervariasi, yang diharapkan dapat menunjukkan kemampuan aplikasi steganography yang dibuat terhadap berbagai macam ukuran citra uji.

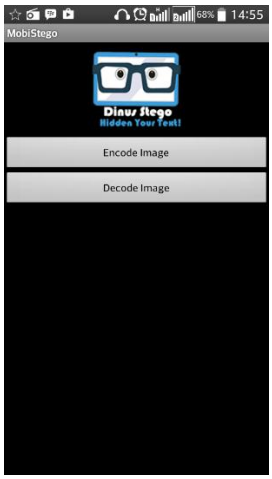
Nama Citra	Gambar Citra	Ukuran Citra (Pixel x Pixel)	Ukuran Citra awal (Kb)
Lena.jpg		329 x 361	255 Kilobyte

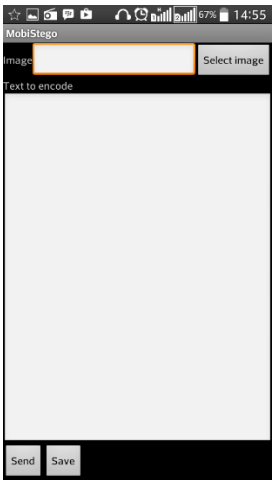
baboon.jpg		638 x 650	76,4 Kb
Semeru.jpg		1600 x 1600	167 Kb
Flower.jpg		1600 x 1600	299 Kb

Analisis Dan Pembahasan

4.1 Proses Penyisipan pesan

Untuk memulai proses penyisipan pesan, pengguna harus membuat berkas gambar cover (JPG) sebagai media penampung pesan (teks) dengan cara mengambil gambar menggunakan kamera smartphone atau gambar yang sudah ada di galeri. Setelah meng-load gambar selanjutnya memasukkan pesan text yang ingin disembunyikan, pada text-area yang sudah tersedia.





Gambar 4.1 -a memilih menu Gambar 4.1 –b Load Image dan Text area.

Setelah memilih file gambar dan mengisi teks, maka proses selanjutnya adalah proses encoding.

Gambar 4.1 -c Proses Encoding Gambar 4.1 –d Proses Saving

Setelah menu encoding akan muncul pilihan menu “save” atau “send”. Pada metode Least Significant Bit, Informasi yang akan di *embedding* dimasukkan ke Bit/ bit. Bit ini akan di masukkan pada bit RGB gambar.

4.2 Pengujian Perangkat Lunak

Tahap pengujian bertujuan untuk memastikan bahwa perangkat aplikasi yang dibuat memenuhi tujuan yang diinginkan. Pada tahap ini perangkat lunak akan diuji berdasarkan parameter yang telah ditetapkan pada batasan masalah, yaitu aspek keamanan data berupa fidelity.

No	Nama Citra	Ukuran Citra Cover	Ukuran Citra Stego	Jumlah Karakter Uji	Hasil Stego
1	Lena.jpg	261byte	238kb	20	72
2	Baboon.jpg	76.4kb	762kb	20	49.4
3	Semeru.jpg	167kb	1,32mb	20	57.7
4	Flower.jpg	299kb	2,68mb	20	48,9
5	Lena.jpg	261byte	238kb	40	72
6	Baboon.jpg	76.4kb	762kb	40	49.4
7	Semeru.jpg	167kb	1,32mb	40	57.7
8	Flower.jpg	299kb	2,68mb	40	48,9
9	Lena.jpg	261byte	238kb	60	72
10	Baboon.jpg	76.4kb	762kb	60	49.4

11	Semeru.jpg	167kb	1,32mb	60	57.7
12	Flower.jpg	299kb	2,68mb	60	48,9
13	Lena.jpg	261byte	238kb	80	62
14	Baboon.jpg	76.4kb	762kb	80	45,45
15	Semeru.jpg	167kb	1,32mb	80	50,7
16	Flower.jpg	299kb	2,68mb	80	43,9
17	Lena.jpg	261byte	238kb	100	72
18	Baboon.jpg	76.4kb	762kb	100	45,4
19	Semeru.jpg	167kb	1,32mb	100	50,7
20	Flower.jpg	299kb	2,68mb	100	43,9
21	Lena.jpg	261byte	238kb	120	70
22	Baboon.jpg	76.4kb	762kb	120	42,4
23	Semeru.jpg	167kb	1,32mb	120	53,7
24	Flower.jpg	299kb	2,68mb	120	41,9
25	Lena.jpg	261byte	238kb	140	70
26	Baboon.jpg	76.4kb	762kb	140	42,4
27	Semeru.jpg	167kb	1,32mb	140	53,7
28	Flower.jpg	299kb	2,68mb	140	41,9
29	Lena.jpg	261byte	238kb	160	70
30	Baboon.jpg	76.4kb	762kb	160	42,4
31	Semeru.jpg	167kb	1,32mb	160	53,7
32	Flower.jpg	299kb	2,68mb	160	41,9

4.4

Hasil Pengujian

PSNR sering dinyatakan dalam skala logaritmik dalam decibel (dB). Nilai PSNR jatuh dibawah 30 dB mengindikasikan qualitas yang relative rendah, dimana distorsi yang dikarenakan penyisipan terlihat jelas. Akan tetapi kualitas stego-image yang tinggi berada pada nilai 40dB dan diatasnya (Cheddad, 2010).

Dari hasil pengujian di atas dapat dilihat bahwa perangkat lunak ini mendukung aspek *fidelity*, artinya JPG hasil steganografi tidak mengalami degradasi yang signifikan dan masih dapat dilihat dengan baik.

Karena pesan yang disisipkan dalam JPG yang dijadikan sebagai penampung pesan tidak mengundang kecurigaan karena tidak mengalami perubahan berarti yang dapat dipersepsi manusia.

Dari Nilai uji PSNR, dapat diketahui pada penyisipan 20-40 karakter ternyata nlai PSNR yang dihasilakan lebih baik dari penyisipan 60-80 karakter, dan uji penyisipan 60-80 karakter ternyata juiga lebih baik dari 100-120 karakter sisipan, dan 100-120karakter uji ternyata nilai

PNSR nya lebih baik dari penyisipan 140-160 karakter uji.

Dari sini dapat disimpuilkan bahwa semakin besar/banyak karakter yang disisipkan akan mempengaruhi kualitas citra yang di bandingkan dengan citra asli(original). Semakin tinggi nilai PSNR nya maka semakin baik, dan Semakin rendah nilai PSNR nya maka akan semakin buruk kualitas citra yang di hasilkan. Dan banyaknya jumlah karakter Uji ternyata tidak berpengaruh pada ukuran citra yang dihasilkan, namun terdapat perbedaan yang cukup terlihat gambar asli dengan citra stego.

5.1 Kesimpulan

Setelah melakukan implementasi dan dilanjutkan pengujian aplikasi, maka dari hasiltersebut dapat ditarik beberapa kesimpulan sebagai berikut :

1. Mengamankan pesan text yang dikirim melalui media digital, dapat disisipkan kedalam gambar sebagai medianya.
2. Perangkat lunak yang mengimplementasi kan steganografi gambar dengan teknik Least Significant Bit pada berkas JPG berhasil dibangun. Kebutuhan fungsional dari perangkat lunak seperti proses penyembunyian dan ekstraksi pesan serta penggunaan kunci sudah dapat dilakukan dengan benar.
3. Penggunaan Kunci dengan teknik Pseudo-random *pseudo number squence*memberikan nilai keamanan lebih karena pesan gambar hanya akan dapat di ekstraksi dengan aplikasi serupa, jadi dapat disimpulkan yang dapat membuka stego-image hanya aplikasi MobiStego.

5.2 Saran

Saran–saran yang dapat diberikan untuk pengembangan lebih lanjut adalah sebagai berikut :

1. Perlu dilakukan pengembangan untuk meningkatkan kapasitas penyembunyian dalam arti penyembunyian bit-bit pesan bisa memanfaatkan celah-celah kosong bit yang belum diisi.

2. Perlu adanya analisis lebih lanjut dan implementasi teknik Least Significant Bit pada format gambar lainnya seperti PNG, GIF, BMP, dll.

Daftar Pustaka

T. Primbada, "Aplikasi Chat dengan Steganografi pada Audio Menggunakan Metode," Aplikasi Chat dengan Steganografi pada Audio Menggunakan Metode, 2013.

E. Cole, "Steganography and the Art of Covert Communication," Steganography and the Art of Covert Communication, 2003.

R.Munir, Kriptografi, Bandung: Informatika, 2006.

W. K. I. Pakka, "Perbandingan Metode Phase Coding Dan Echo Data Hiding Dengan Keamanan Data Menggunakan Algoritma Rijndael Pada Steganografi Berkas Audio," Perbandingan Metode Phase Coding Dan Echo Data Hiding Dengan Keamanan Data Menggunakan Algoritma Rijndael Pada Steganografi Berkas Audio, 2010.

Hermawan, B. 2004. Menguasai Java 2 dan Object Oriented Programming. Yogyakarta: Andi..

Prahasta, E., 2002, System informasi Geografi tutorial ArcView, Informatika Bandung, Bandung.

Pramadya, J. S. A. 2011. Pembuatan Aplikasi Mobile Berbasis Android Os Untuk Mengetahui Lokasi Tempat Wisata Di Daerah Istimewa Yogyakarta (Skripsi). Yogyakarta: Stmik Amikom.

Safaat, N. 2010. Membangun Aplikasi Mobile Berbasis Android. Bandung: Informatika Bandung

Saputro, S. 2008. Pengelolaan Database MySQL dengan PhpMyAdmin. Yogyakarta: Graha Ilmu.

Shalahuddin & Rosa. 2008. Java di Web. Bandung: Informatika Bandung

Wijaya, A. J., Stefanie, and Ruthy, M. 2011. Aplikasi Promosi Dan Pencarian Rumah Tinggal Berbasis Android (Kerja Praktek). Jakarta: FTI Universitas Binus

Yuhri, T. 2011. Membangun Aplikasi Web Potensi Wisata Dan Kuliner Tingkat Kecamatan Srandakan Di Kabupaten Bantul Berbasis Sistem Informasi Geografis (Kerja Praktek). Yogyakarta: FST IST Akprind

Google, Inc., 12 Oktober 2012, Android Developers Documentation, developer.android.com/index.html