# Kriptografi Dan Kompresi Pesan Singkat Pada Android

**Pizaini[1], Febi Yanto[2]**
Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim, Riau
Jl. HR Soebrantas No. 155 KM 15. Pekanbaru, Riau
e-mail: pizaini@yahoo.com, ebied91@yahoo.com

***Abstrak***

*Komunikasi melalui Short Message Service (SMS) merupakan teknologi pengiriman pesan yang masih memiliki beberapa permasalahan yaitu ukuran pesan yang terbatas dan keamanan isi pesan. Ukuran pesan yang lebih dari 140 byte akan dikirim lebih dari satu SMS dan isi pesan dapat dibaca oleh orang lain jika memiliki akses ke SMS Center. Implementasi kriptografi dan kompresi pada SMS dapat meningkatkan keamanan pesan serta meningkatkan efisiensi jumlah pengiriman pesan. Algoritma Advanced Encryption Standard (AES) merupakan salah satu metode kriptografi yang handal untuk mengamankan data dan dikombinasikan dengan kompresi Huffman yang menggunakan prinsip statistik untuk menghasilkan data yang berukuran lebih kecil. Kombinasi enkripsi dan kompresi dapat menghasilkan empat mode pengiriman pesan yaitu Compress Only, Compress and Encrypt, Encrypt and Compress dan Encrypt Only. Mode pengiriman Compress Only dapat mengurangi ukuran pesan dengan rasio kompresi sebesar 72,87 %. Sedangkan mode pengiriman Encrypt Only dapat meningkatkan keamanan pesan, tetapi menghasilkan pesan yang lebih besar dari pesan asli. Jika kompresi dan enkripsi dikombinasikan menghasilakan rasio kompresi sebesar 75,94 %.*
***Kata kunci:****AES, Enkripsi, Huffman, Kompresi, SMS*

***Abstract***

*Communication via Short Message Service (SMS) messaging is a technology that still has some problems that limited message size and message content security. Message size is greater than 140 bytes are sent over the SMS and the message can be read by others if they have access to the SMS Center. Implementation of cryptographic and compression on the SMS message can increase security and improve the efficiency of the delivery of the message. Algorithm Advanced Encryption Standard (AES) is one of the reliable cryptographic methods to secure data and combined with Huffman compression that uses statistical principles to produce data that are smaller. The combination of encryption and compression can result in four modes, namely messaging Only Compress, Compress and Encrypt, Encrypt and Compress and Encrypt Only. Only Compress delivery mode can reduce the size of the message with a compression ratio of 72.87%. Only while the Encrypt transmission mode can improve the security of a message, but it produces a larger message of the original message. If the combined compression and encryption menghasilakan compression ratio of 75.94%.*

***Keywords****: AES, Encryption, Huffman, Compression, SMS*

## 1. Pendahuluan

Perkembangan *mobile phone* saat ini dapat menggambarkan besarnya animo masyarakat dalam perkembangan teknologi. Selain komunikasi suara, fasilitas *Short Message Service* (SMS) yang disediakan dapat digunakan untuk pengiriman pesan singkat yang penggunaannya mudah dan dengan *cost* yang lebih murah dibandingkan dengan komunikasi suara.

Pengiriman pesan dengan SMS masih memiliki permasalahan di antaranya adalah ukuran pesan yang akan dikirim sering berukuran besar sehingga memakan waktu yang cukup lama dalam proses transmisi data tersebut. Selain itu dalam penyimpanan data, pesan yang cukup besar memakan ruang yang besar pula. SMS tersebut memiliki batasan 160 karakter (karakter GSM 7 bit), jika pesan yang akan dikirim lebih dari 160 karakter, maka *mobile phone* akan mengirim dua SMS meskipun pesan tersebut hanya satu karakter lebih besar dari batasan satu pesan [2].

Beberapa cara dapat dilakukan untuk mengatasi hal tersebut, salah satunya adalah dengan melakukan kompresi. Kompresi merupakan proses *encoding* data menggunakan jumlah bit yang lebih kecil, sehingga bit yang lebih kecil tersebut dapat merepresentasikan informasi yang sama. Kompresi teks atau data akan memperkecil jumlah memori yang digunakan dan mempercepat tercapainya proses transmisi pesan bahkan dapat menekan biaya pengiriman.

Selain memperkeci lukuran pesan untuk mempercepat proses transmisi data, keamanan pesan juga perlu diperhatikan. Celah keamanan terbesar pada komunikasi via SMS adalah pesan yang dikirimkan akan disimpan di *Short Message Service Center* (SMSC), yaitu tempat dimana SMS disimpan sebelum dikirim ke tujuan. Pesan yang sifatnya *plaintext* ini dapat disadap dan dibaca oleh siapa saja

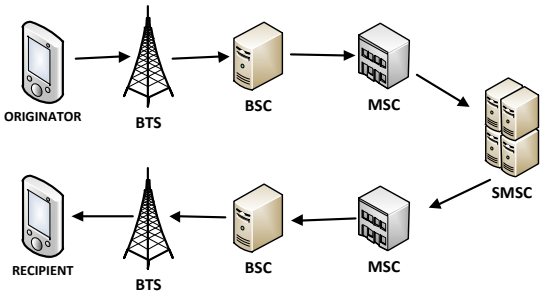
yang memiliki akses ke dalam SMSC. Akibatnya, informasi penting dapat dibaca oleh orang yang tidak berhak untuk mengetahuinya [8].

Aplikasi enkripsi dan dekripsi pesan dapat meningkatkan keamanan pada layanan yang memerlukan kerahasiaan pesan. Hal ini dapat mengurangi bocornya informasi kepada pihak-pihak yang tidak berkepentingan seperti operator telepon seluler. Untuk meningkatkan keamanan dalam aplikasi enkripsi dan dekripsi pesan SMS, perlu digunakan algoritma yang handal. Salah satu algoritma yang handal adalah algoritma *Advanced Encryption Standard* (AES).

## 2. Landasan Teori

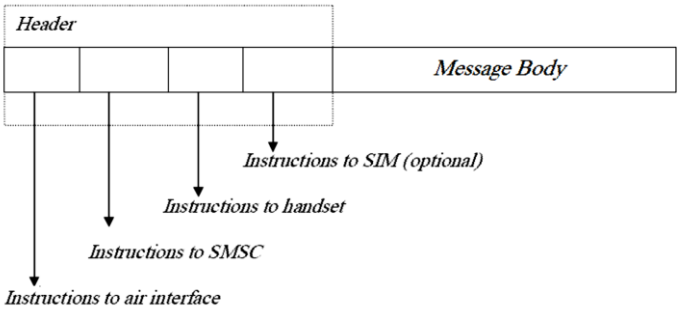
### Short Message Service (SMS)

*Short Message Service* (SMS) adalah kemampuan untuk mengirim dan menerima pesan dalam bentuk teks dari dan kepada ponsel. Teks tersebut bisa terdiri dari kata-kata atau nomor atau kombinasi alphanumeric. SMS diciptakan sebagai standar pesan (*message*) oleh ETSI (*Europesan Telecommunication Standards Institute*), yang juga membuat standar GSM yang diimplementasikan oleh semua operator GSM (saat ini standar SMS menjadi tanggung jawab 3GPP - *Third Generation Partnership Project*)




Gambar 1. Mekanisme Pengiriman SMS

Pada sebuah paket pesan SMS terdiri dari *header* dan *body*. *Header* pesan terdiri dari instruksi-instruksi kepada komponen-komponen yang bekerja dalam jaringan SMS. Pada instruksi-instruksi tersebut, terdapat informasi yang diperlukan selama pengiriman pesan seperti informasi validitas pesan, dan informasi-informasi lainnya. Pada bagian *message body*, terdapat isi dari pengirim pesan yang akan dikirimkan. Berikut gambar Struktur Data dari SMS:




Gambar 2. Struktur *Short Message Service* (SMS)

### AES (*Advanced Ecrypytion Standard*)

AES dipublikasikan oleh NIST (*National Institute of Standard and Technology*) pada tahun 2001. AES merupakan blok kode simetris untuk menggantikan DES (*Data Ecncription Standard*).

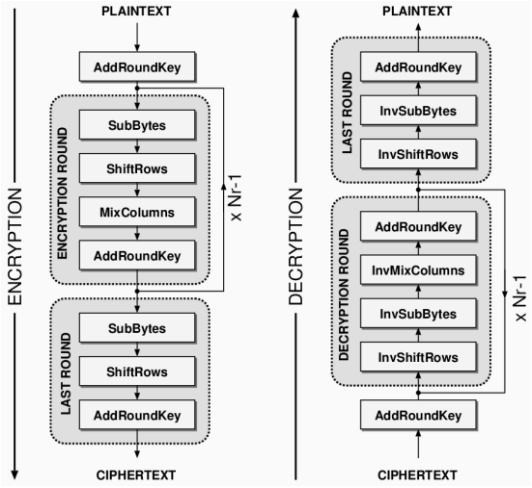
Tabel 1. Parameter AES

| - | *AES-128* | *AES-192* | *AES-256* |
|---|---|---|---|
| *Key Size* | 16 *byte* | 24 *byte* | 32 *byte* |
| *Block Size* | 16 *byte* | 16 *byte* | 16 *byte* |
| *Round* | 10 | 12 | 14 |
| *Round Key* | 16 *byte* | 16 *byte* | 16 *byte* |

Blok masukan akan diproses dalam bentuk *array* dan menghasilkan *ciphertext*. Algoritma AES memiliki empat transformasi utama yaitu [1]:

a. *SubBytes* adalah substitusi *byte* dengan menggunakan tabel substitusi (*S-Box*).

b. *ShiftRows* adalah pergeseran baris-baris *array state* secara *wrapping*.
c. *MixColumns* adalah mengacak data di masing-masing kolom *array state.*
d. *AddRoundKey* adalah melakukan *XOR* antara state sekarang dengan *round key.*




Gambar 3. Proses Enkripsi dan Dekripsi AES

Garis besar algoritma AES yang beroperasi pada blok 128 bit adalah sebagai berikut:
*a. AddRoundKey*, melakukan *XOR* antara *state* awal (*plaintext*) dengan *chiper key*. Tahap ini disebut juga *initial round.*
b. Putaran sebanyak Nr–1 kali. Proses yang dilakukan pada setiap putaran adalah *SubBytes, ShiftRows, MixColumns* dan *AddRoundKey.*
*c. Final round*, proses untuk putaran terakhir yaitu*SubBytes, ShiftRows* dan *AddRoundKey.*

***Huffman***

Salah satu teori yang dapat digunakan untuk mengkompresi data adalah dengan kode Huffman. Kode ini dikemukakan oleh David A. Huffman pada tahun 1952. Dalam kompresi data, kode Huffman adalah kode-kode biner yang mengodekan suatu simbol tertentu pada suatu data. Kode-kode tersebut dibentuk dengan memperhatikan frekuensi kemunculan simbol tertentu pada data tersebut.

Kode Huffman menerapkan konsep kode awalan (*prefix code*), yang merupakan himpunan kode biner, sedemikian sehingga tidak ada anggota himpunan yang merupakan awalan dari anggota yang lain, supaya pada proses dekoding, tidak ada keambiguan antara satu simbol dengan simbol yang lain.

Kode awalan yang merepresentasikan simbol yang lebih sering muncul menggunakan rangkaian biner yang lebih pendek dari pada kode yang digunakan untuk merepresentasikan simbol yang lebih jarang muncul. Dengan demikian jumlah bit yang digunakan untuk menyimpan informasi pada suatu data bisa lebih pendek.

**3. Hasil dan Analisis**

Proses enkripsi pesan singkat (SMS) pada dasarnya sama seperti proses enkripsi data lainnya. Pada enkripsi SMS perlu dilakukan konversi data antara karakter GSM ke biner dan sebaliknya, karena pesan yang akan dikirim dalam format karakter GSM *Default 7 bit*.

*Advanced Encryption Standard* (AES) merupakan salah satu algoritma kriptografi yang bersifat *block chiper*. Artinya, proses enkripsi dan dekripsi dilakukan pembagian blok berukuran 16 *byte* (128 bit). Pesan yang akan dienkripsi dimasukkan pada blok tersebut. Jika pesan lebih dari 128 bit, pesan tersebut akan diproses setelah proses 16 *byte* pertama selesai. Pesan yang akan dienkripsi tidak selalu berukuran kelipatan 128 bit. Oleh karena itu, jika ukuran blok kurang dari 128 bit, maka harus ada karakter tambahan agar bisa memenuhi jumlah 128 bit tersebut (*padding*).

Komporesi dilakukan berdasarkan Tabel Huffman. Tabel tersebut akan digunakan sebagai acuan untuk melakukan proses kompresi. Tabel Huffman dibuat dengan cara menentukan kode Huffman dari karakter-karakter *Standard* GSM 03.38. Penentuan kode Huffman ini dilakukan dengan cara memberi kode yang pendek untuk karakter – karakter SMS tersebut

Pesan yang akan dikirim terlebih dahulu diproses untuk dikompresi dengan membaca seluruh karakter yang dimasukkan. Kemudian dilakukan proses enkoding masing-masing karakter berdasarkan tabel Huffman yang telah dibangun. Hasil enkoding ini kemudian dikonversi ke karakter GSM 7 bit yang akan dikirim sebagai SMS. Proses pengiriman pesan secara umum adalah:
1. Pengguna mengetikkan pesan yang akan dikirim ke pengguna lainnya.
2. Aplikasi akan mengambil data SMS atau pesan yang akan dikirim tersebut.

3. Pesan tersebut adalah karakter GSM yang kemudian di-enkoding berdasarkan tabel statis yang telah ada. Setiap karakter akan di-enkoding hingga seluruh karakter pesan selesai.
4. Hasil dari enkoding ini adalah pesan yang terkompresi yang berbentuk bilangan biner.
5. Pesan biner inilah yang akan dikonversi dan dikirim ke penerima dalam bentuk karakter GSM *Default* 7 bit.

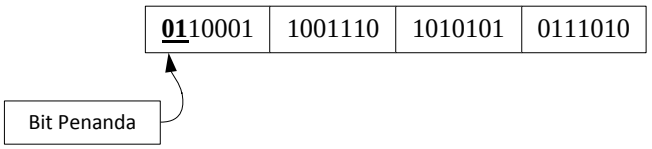
Pesan yang terkirim akan diterima oleh penerima dengan langkah-langkah sebagai berikut:

1. Pesan yang terima dalam bentuk teks karakter *Default* GSM akan diterjemahkan kembali menjadi pesan aslinya.
2. Ambil seluruh pesan yang diterima dalam format bilangan biner.
3. Lakukan dekoding bilangan tersebut berdasarkan tabel statik yang telah ada.
4. Hasil dari dekoding (dekompresi) ini adalah karakter GSM, sehingga tidak perlu dilakukan konversi data dan pesan ini dapat langsung ditampilkan ke penerima.

Bit penanda berfungsi agar aplikasi di perangkat penerima pesan mengetahui mode pengiriman yang digunakan. Bit penanda ini adalah:

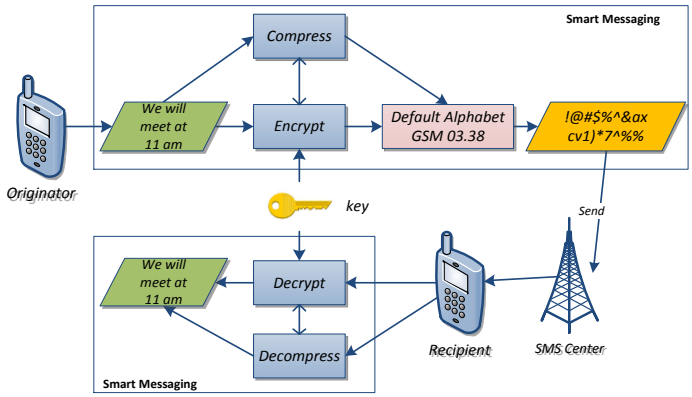
1. 00 untuk pengiriman kompresi
2. 01 untuk pengiriman kompresi dan enkripsi
3. 10 untuk pengiriman enkripsi dan kompresi
4. 11 untuk pengiriman enkripsi

Misalnya, pesan yang akan dikirim (sudah dalam bentuk biner) adalah 10001100111010101010111010, maka untuk pengiriman dengan *mode Compress and Encrypt* adalah dengan menambahkan bit 01 di awal pesan. Maka pesan akan menjadi **01**10001100111010101010111010.




Gambar 4. Penambahan Bit Penanda pada Pesan

Perancangan Aplikasi ini adalah sebagai berikut:

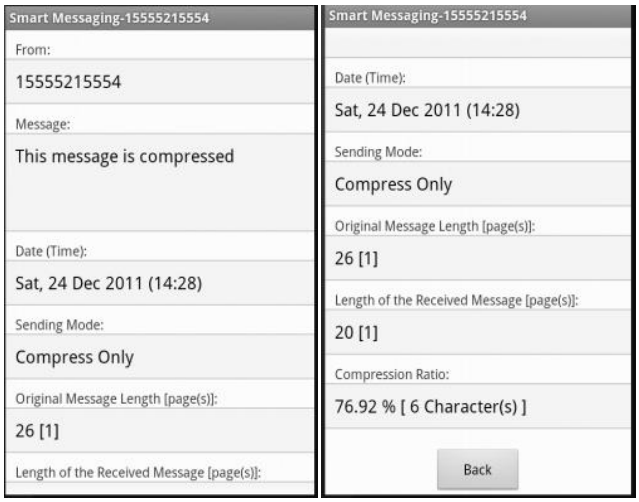



Gambar 5. Perancangan Aplikasi

Berdasarkan Gambar 5 dapat dijelaskan bahwa aplikasi ini berfungsi untuk meng-enkripsi dan mengkompresi pesan. Aplikasi ini harus berada di kedua sisi pengirim dan penerima pesan. Pesan yang diketikkan pengguna pada sisi pengirim berbentuk pesan singkat pada umunya yaitu menggunakan karakter *Default* GSM. Ketika pesan tersebut diproses oleh aplikasi, pesan ini akan di encoding baik kompresi maupun enkripsi. *Output* dari aplikasi ini adalah pesan yang berbentuk *Default* GSM dan akan dikirim sebagai SMS.

Penerima (*recipient*) akan menerima pesan dari *Short Message Service Center* (SMSC) dalam bentuk karakter GSM *Default*. Aplikasi akan mengenali mode pengiriman yang digunakan kemudian melakukan dekompresi dan dekripsi. Pesan kemudian dikonversi menjadi kode GSM *Default* dan ditampilkan ke penerima.

*ISSN : 2085-9902*



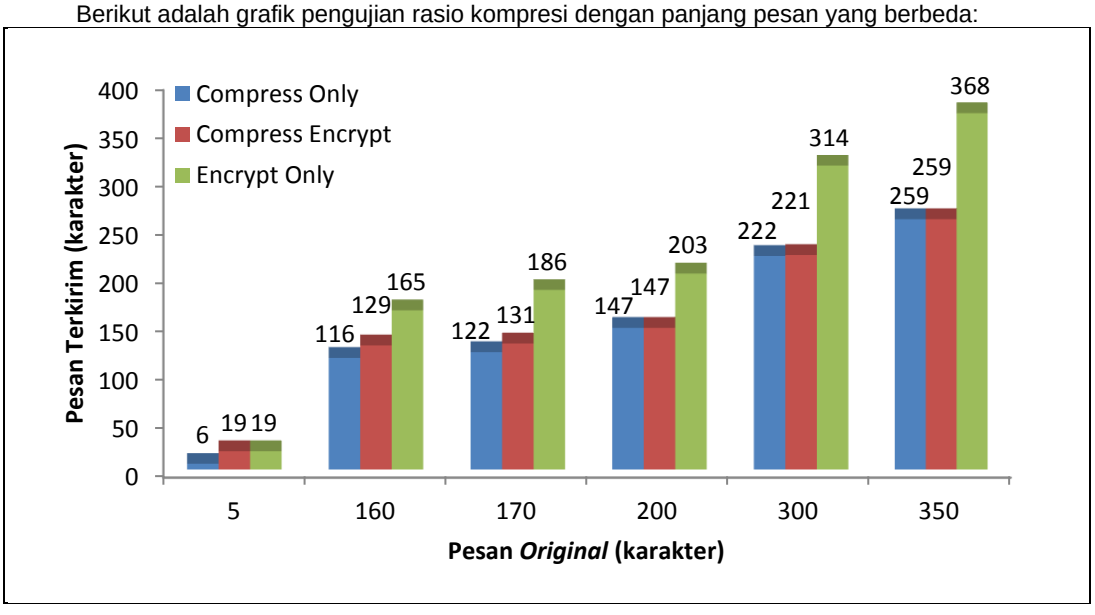

Gambar 6. Hasil Implementasi Detail Pesan

Tabel 2. Pengujuan Rasio Kompresi

| Mode Pengiriman | Panjang bit *before* | Panjang bit *after* | Panjang Pesan dikirim | Rasio Karakter | Jumlah Halaman |
|---|---|---|---|---|---|
| *Compress Only* | 1190 bit | 843 bit | 122 karakter | 71,76 % | 1 |
| *Compress Encrypt* | 1190 bit | *Compress* = 843, *Encrypt* = 896 | 130 karakter | 76,47 % | 1 |
| *Encrypt Compress* | 1190 bit | *Encrypt* = 1280, *Compress* = 758 | 112 karakter | 65,88 % | 1 |
| *Encrypt Only* | 1190 bit | 1280 bit | 184 karakter | 108,23 % | 2 |

Tabel 2 menjelaskan pengujian rasio kompresi pada empat mode pengiriman pesan. Misalnya pada mode Compress Only dengan jumlah 170 karakter GSM (1190 bit). Setelah proses kompresi, pesan menjadi 843 bit. Sehingga pesan yang akan dikirim adalah 122 karakter. Pada mode tersebut rasio komprei adalah 71,76 % dan jumlah pesan menjadi satu halaman. Pada pengujian ini, mode Encrypt Encrypt tidak dapat mengemblikan pesan asli. Hal ini karena terbatasnya jumlah karakter dalam table Huffman, sehingga karakter yang tidak terdapat dalam tabel Huffman tidak dapat dikompresi (diabaikan).

Berikut adalah grafik pengujian rasio kompresi dengan panjang pesan yang berbeda:




Gambar 7. Grafik Pengujian Rasio Kompresi

Gambar 7 menjelaskan hasil pengujian beberapa variasi panjang pesan yang dikirim. Berdasarkan gambar tersebut, pengujian mode Encrypt Only menghasilkan pesan yang lebih panjang dari pesan aslinya. Sedangkan mode Compress Ony dan Compress and Encrypt dapat mengurangi karakter pesan yang akan dikirim.

**4. Kesimpulan dan Saran**

1. Berdasarkan hasil pengujian pada bab sebelumnya, mode pengiriman *Compress Only* dan *Compress Encrypt* dapat menghasilkan kompresi yang baik dengan rasio kompresi rata-rata 72,87 % dan 75,94 %. Pada mode *Encrypt Only* menghasilkan pesan yang lebih besar dari pesan aslinya dengan rasio kompresi hingga 110 %.
2. Mode pengiriman *Compress Only* dan *Compress Encrypt* secara efektif dapat mengurangi jumlah halaman pesan dengan panjang pesan hingga 200 karakter (2 halaman) menjadi satu halaman pesan, serta panjang pesan hingga 350 karakter (3 halaman) menjadi dua halaman pesan.
3. Hasil pengujian untuk panjang pesan kurang dari 10 karakter, menghasilkan rasio kompresi lebih dari 100 %. Artinya, jika pengirim memasukkan karakter yang sedikit, seluruh mode pengiriman menghasilkan pesan yang lebih besar dari pesan aslinya.
4. Pada penelitian ini menggunakan metode kompresi Huffman dan tabel Huffman statis. Berdasarkan hasil pengujian, bahwa rasio kompresi masih berukuran besar. Oleh karena itu, disarankan kompresi pesan tidak hanya menggunakan satu metode saja tetapi dapat dikombinasikan dengan metode kompresi data lainnya.

**Referensi**

[1] Ariyus, Dony. PengantarIlmuKriptografi: Teori, Analisis, danImplementasi. Yogyakarta:Andi. 2008: 13, 85, 165 – 182
[2] Citra, Anatasya. "*Aplikasi Kompresi Sms Menggunakan Kode Huffman Pada Mobile Phone Berbasis Java*", 2010,http://repository.usu.ac.id/handle/123456789/19371
[3] Gunawan, Ferry. Membuat Aplikasi SMS Gateway dan Client dengan Java dan PHP. Jakarta:Elex Media Komputindo. 2003
[4] Indarto, Wawandkk. Implementasi Algoritma Run Length, Half Byte Dan Huffman Untuk Kompresi File. SNATI. 2005; 979-756-061-6
[5] Linawati dan Panggabean, Henry P. Perbandingan Kinerja Algoritma Kompresi Huffman, LZW, dan DMC pada Berbagai Tipe File. Integral. 2004; Vol. 9 No. 1
[6] Safaat. H, Nazruddin. PemrogramanAplikasi Mobile Smartphone dan Tablet Berbasis Android. Bandung: Informatika. 2011
[7] Prabawa, I.Y.B. Aditya Eka. "*Kompresi Data dengan Kode Huffman dan Variasinya*", 2008, http://repository.usu.ac.id/bitstream/123456789/14093/1/ 09E01151.pdf
[8] Prasetyo, Galih Wahyu."*Aplikasi Enkripsi SMS Menggunakan Metode Blowfish*", 2010, http://repo.eepis-its.edu/723/1/1021.pdf (5 September 2011)
[9] Putra, Darma. Pengolahan Citra Digital. Yogyakarta : Andi. 2010: 261 – 270
[10] Sofwan, Aghus dkk. Aplikasi Kriptografi Dengan Algoritma Message Digest 5 (MD5). Transmisi.2006; Vol. 11 No. 1 : 22 - 27