Aplikasi *Watermarking* Citra Digital Berbasis *Mobile phone*

I Nyoman Prama Pradnyana - 13509032
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
prama.pradnyana@itb.ac.id

***Abstract*—Saat ini, hampir seluruh *mobile phone* yang ada menyediakan berbagai fasilitas tambahan yang membantu pengguna seperti fasilitas pengambilan citra pada *mobile phone*. Akhir-akhir ini marak terjadi penyalahgunaan fasilitas tersebut oleh pihak yang tidak bertanggungjawab. Penyalahgunaan yang terjadi antara lain berbentuk manipulasi citra seperti *cropping* dll. Hal ini marak terjadi mengingat *mobile phone* merupakan salah satu perangkat yang paling praktis dan sering digunakan dalam mengambil suatu citra. Hasil manipulasi ini kemudian disebarkan ke berbagai *mobile phone* lain. Pengguna lain yang menerima gambar tersebut tentu saja tidak mengetahui bahwa gambar tersebut merupakan hasil manipulasi. Permasalahan yang terjadi adalah akan ada pihak-pihak yang dirugikan dari penyalahgunaan ini. Salah satu cara untuk membantu mengatasi permasalahan tersebut adalah dengan memberikan suatu penanda pada gambar. *Watermarking* merupakan salah satu metode pemberi tanda pada suatu citra digital. Pada makalah ini akan dijelaskan pengimplementasian *watermarking* pada perangkat *mobile phone* dengan menggunakan platform J2ME dan pengecekan terhadap kualitas gambar yang dihasilkan. Dari hasil implementasi dan dilakukan perhitungan PSNR didapatkan bahwa pada *mobile phone*, penggunaan *watermark*ing bisa dilakukan tanpa merusak gambar.**

Index Terms- J2ME, Watermarking, PSNR

I. PENDAHULUAN

Saat ini, perkembangan teknologi sudah sangat pesat. Teknologi sudah menjadi salah satu kebutuhan bagi masyarakat. Salah satu bentuk teknologi yang sudah merambah ke dalam masyarakat adalah *mobile phone* atau sering disebut dengan *handphone*. *Mobile phone* sudah menjadi kebutuhan bagi masyarakat. Tidak hanya golongan atas saja yang menggunakan *mobile phone* namun hampir seluruh masyarakat telah menggunakan *mobile phone*.

Salah satu fitur yang terdapat pada *mobile phone* adalah fitur pengambilan gambar. Hampir seluruh *mobile phone* sudah memiliki fitur ini. Dengan fitur ini, pengguna dapat mengabadikan gambar hanya dengan menggunakan *mobile phone*. Disamping itu *mobile phone* menyediakan fasilitas pengiriman data. Hal ini pengguna dapat melakukan pertukaran data seperti citra digital dll.

Pada awalnya tidak terjadi permasalahan dalam melakukan pengiriman data. Namun seiring dengan berjalannya waktu, terjadi penyalahgunaan fasilitas tersebut. Sangat sering terjadi manipulasi gambar dari citra *mobile phone* dan disebarkan ke berbagai *mobile phone*. Hal ini sudah sangat sering terjadi mengingat sangat mudahnya pengguna untuk mengambil suatu gambar dan mudah untuk disebarkan. Pihak yang tidak bertanggungjawab akan melakukan manipulasi terhadap citra tersbut misalkan dilakukan *cropping* pada gambar. Hal ini bisa merugikan suatu pihak yang menjadi korban manipulasi gambar. Pengguna lain juga tidak mengetahui apakah gambar tersebut asli atau bukan.

Untuk itu dirasa perlu adanya fasilitas pemberi tanda pada suatu citra digital. *Watermarking* adalah salah satu metode yang bisa diterapkan sebagai penanda suatu citra digital. Jika suatu citra disisipkan suatu *watermark*, maka citra tersbut akan tertanda sebagai citra asli. Jika citra tersebut dirubah dengan melakukan manipulasi secara langsung terhadap citra tersebut, data citra tersebut akan berubah. Dengan perubahan data pada citra digital tersebut, pemilik asli dapat menunjukkan data yang asli dan menunjukkan bahwa citra lain merupakan citra hasil manipulasi.

Pada makalah ini akan dibahas mengenai pembuatan aplikasi *watermark*. Pembuatan aplikasi ini menggunakan platform J2ME dan hasil dari pembuatan aplikasi tersebut akan diimplementasikan pada perangkat *mobile phone*. Dengan demikian citra pada *mobile phone* akan memiliki data digital sebagai penanda citra asli. Disamping itu akan dilakukan pengecekan terhadap derajat PSNR dimana derajat ini akan menunjukkan kualitas gambar yang dihasilkan setelah dilakukan penyisipan *watermark* ke dalam citra tersebut.

Melalui makalah ini diharapkan mampu memberikan

gambaran mengenai pembuatan aplikasi *watermark* dan dapat menjadi solusi dari permasalahan penyalahgunaan fasilitas *mobile phone*.

II. DASAR TEORI

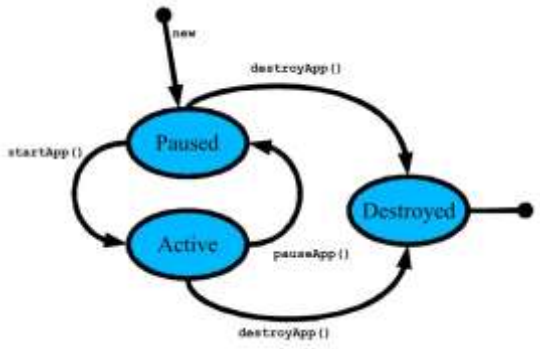
2.1 J2ME

J2ME adalah teknologi java yang diperuntukkan bagi perangkat-perangkat kecil seperti consumer device, terutama wireless. Beberapa perangkat wireless antara lain PC, PDA, dan ponsel. Pada J2ME ini terdapat beberapa jenis platform antara lain JVM (Java Virtual Machine), KVM dan Card VM. Pada pembuatan aplikasi ini akan digunakan platform KVM yaitu MIDlet. Hal ini dikarenakan MIDlet sangat cocok untuk *mobile phone*.

MIDlet merupakan aplikasi yang dijalankan pada sebuah perangkat *handled*. MIDlet tidak berinteraksi langsung dengan *hardware* dari *handheld devices*, melainkan berinteraksi melalui AMS (*Application Management Software*). AMS inilah yang akan menerima sinyal dari MIDlet bahwa MIDlet akan dijalakan atau berhenti.

Midlet dapat dibagi menjadi tiga berdasarkan prosesnya ketika dijalankan. Ketiga proses tersebut antara lain :

- Paused, Status ini terjadi ketika MIDlet selesai disosialisasikan dan tidak melakukan aksi apapun.
- Active, Status ini terjadi ketika MIDlet sedang berjalan dengan normal, yakni setelah memanggil fungsi MIDlet.startApp()
- Destroyed, Status ini terjadi ketika MIDlet berhenti berjalan (exit), sehingga seluruh sumber daya yang digunakan akan dibebaskan. Status ini terjadi ketika berhasil dilakukan pemanggilan fungsi MIDlet.destroyApp() atau MIDlet.notifyDestroyed().



Gambar 2.1 Proses J2ME

2.2 *Watermarking*

Watermarking adalah teknik untuk menyisipkan informasi tertentu ke dalam data digital. Data digital yang disisipkan disebut dengan *watermark*. *Watermarking* merupakan bentuk aplikasi dari steganografi. Pada prosesnya, *watermark* dapat melakukan penyisipan teks, citra, suara, maupun video ke dalam data digital. Hal ini biasanya dilakukan untuk memberi perlindungan *copyright* terhadap suatu data digital. Jika data digital tersebut tersebar, data *watermark* akan tetap berada pada data digital tersebut sehingga ketika terjadi pengakuan terhadap data digital tersebut, pemilik asli dapat menunjukkan *watermark* yang ada pada data digital tersebut dan menunjukkan bahwa data digital tersebut adalah miliknya.

Gambar 2.2 *Watermark*ing

Pada proses *watermarking* ini akan digunakan tiga data penting yaitu data digital asli, item yang akan disisipkan ke dalam data digital asli, dan kunci. Data digital asli dan item dapat berupa gambar, musik ataupun video. Sedangkan kunci berupa teks singkat. Kunci ini digunakan untuk mencegah penghapusan secara langsung oleh pihak tak bertanggungjawab, namun ketahanan terhadap proses pengolahan lainnya tetap bergantung kepada metode *watermark*ing yang digunakan.

Proses *watermaking* adalah melakukan penyisipan item ke dalam citra asli. Penyisipan dilakukan dengan menyisipkan tiap bit item yang disisipkan ke dalam byte gambar asli. Terdapat beberapa metode yang bisa digunakan dalam melakukan proses penyisipan salah satunya adalah metode spatial domain yang terdiri dari metode RSB dan LSB. Pada pembuatan program ini akan digunakan metode LSB mengingat keterbatasan dari perangkat *mobile phone*.

Gambar 2.3 RSB dan LSB

Metode LSB (Least Significant Bit) ini merupakan metode yang memanfaatkan kelemahan indra manusia. *Watermark* yang menggunakan metode LSB adalah melakukan penggantian bit LSB piksel dengan bit data.

Pengubahan bit ini tentu akan menyebabkan perubahan pada citra digital yang disisipi. Namun pada kenyataanya perubahan nilai LSB ini masih tidak dapat ditangkap oleh indra manusia. Oleh karena itu metode LSB sering digunakan. Meskipun LSB sangat praktis dan tidak merusak citra, LSB memilki beberapa kelemahan salah satunya metode LSB ini adalah tidak kokoh dalam perubahan. Maksudnya, ketika suatu citra telah selesai disispi *watermark* dengan menggunakan LSB, maka LSB tersebut akan hilang ketika citra tersebut diberikan *watermark* lain dengan menggunakan LSB. Jika hal ini terjadi maka data hasil *watermark* itu akan terhapus atau tergantikan dengan *watermark* yang baru.

2.3 PSNR

PSNR (*Peak Signal to Noise Ratio*) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya kerusakan yang mempengaruhi sinyal tersebut. Pada makalah ini, PSNR akan digunakan sebagai pengukur kualitas gambar setelah dilakukan penyisipan. Perumusan PSNR yang digunakan pada makalah ini dapat dituliskan sebagai berikut :

$$PSNR = 20 \times \log_{10}\left(\frac{256}{rms}\right)$$

$$rms = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_{ij} - \hat{I}_{ij})^2}$$

Arti perumusan di atas adalah 256 menunjukkan nilai sinyal terbesar (pada citra dengan 256 derajat keabuan). Sedangkan rms adalah akar pangkat dua dari kuadrat selisih dua buah citra I dan I' yang berukuran MxN. Jika RMS bernilai kecil berarti dua buah citra tersebut mempunyai perbedaan yang sedikit. Sedangkan jika dilihat dari hasil PSNRnya, PSNR memiliki arti jika PSNR>30, citra tersebut masih dikatakan berkualitas bagus. Sedangkan jika PSNR<30 maka citra tersebut dikatakan sudah terdegradasi

III. METODE DAN IMPLEMENTASI

3.1 Persiapan MIDlet J2ME

Pada pembuatan aplikasi MIDlet ini, digunakan aplikasi pembantu yaitu NetBeans IDE 7.0.1. Pada tahap ini dibuat tahapan penggunaan program dan hasil implementasi di salah satu *mobile phone*.

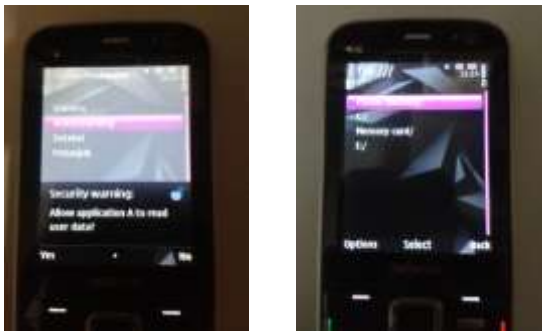
Pembuatan MIDlet ini sesuai dengan proses MIDlet secara umum yang terdiri dari tiga bagian yaitu start, pause, dan destroy. Pada tahap ini dibuat alur yang akan digunakan oleh pengguna dalam menggunakan program. Setelah dilakukan pengkodean dengan membuka *file*system dari *mobile phone*, kompilasi tidak lagi dilakukan pada aplikasi NetBeans karena *file*system yang tersedia pada NetBeans berbeda dengan *mobile phone*. Pada saat ini langsung dilakukan pengecekan dengan mengirimkan .jar program ke *mobile phone* dan langusng dijalankan.

Hasil dari pembuatan program langsung diimplementasikan ke dalam *mobile phone*. Aplikasi ini menyediakan empat buah pilihan yaitu Camera, *Watermark*ing, Deteksi dan Petunjuk. Pada proses implementasi ke *mobile phone*, didapat hasil sebagai berikut:

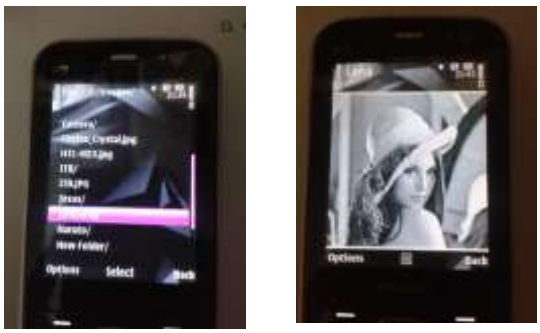
Gambar 3.1 Implementasi pada Nokia N78

Kamera adalah menu yang digunakan untuk mengambil gambar dengan menggunakan kamera. Nantinya gambar ini akan disimpan pada direktori yang sama dengan default penyimpanan hasil kamera *mobile phone*. Pada pembuatan program ini, Kamera tidak diimplementasikan karena setelah dilakukan uji coba, hasil pengambilan citra melalui kamera selalu menghasilkan tipe data JPG/JPEG. Sedangkan program yang akan dibuat adalah untuk data yang berekstensi bitmap (.bmp). JPG tidak bisa dilakukan proses pada *watermark* kali ini karena JPG merupakan tipe data yang sudah terkompres dan tidak bisa diganti bytenya secara acak.

*Watermark*ing adalah menu yang digunakan untuk melakukan *watermarking*. Program akan meminta pengguna untuk memilih *file* dengan melakukan pencarian terhadap *file system*. Setelah memilih *file* citra, kemudian program akan meminta masukkan teks yang akan disisipkan ke dalam citra yang telah dipilih. Pengguna dapat mengetikkan pada label yang tersedia. Setelah memasukkan teks yang akan disisipkan, program akan meminta masukkan kunci. Kunci yang dimaksudkan disini adalah kunci berupa angka 1-100. Setelah memasukkan kunci maka program akan menampilkan pesan dan menyimpan citra hasil penyisipan dengan nama dan path yang sama dengan gambar awal. Pada tahap *watermark*ing ini belum semua fungsi terimplementasi dengan baik.



Gambar 3.2 Membuka *FileSystem*



Gambar 3.2 Membuka *file bitmap*

Gambar 3.3 Meminta masukkan

Gambar 3.4 Setelah *watermark*ing

Untuk menu deteksi, program akan meminta pengguna untuk memilih gambar yang ingin dilihat *watermark*-nya. Tahap ini sama dengan proses *watermark* yaitu memilih gambar dengan menggunakan *filesystem*. Selanjutnya program akan meminta pengguna untuk memasukkan kunci. Jika kunci yang dimasukkan benar, maka akan keluar pesan yang tersisip pada gambar tersebut, namun jika salah maka tetap akan keluar pesan namun pesan yang keluar merupakan pesan tak berarti.

Yang terakhir adalah petunjuk adalah menu yang dapat dipilih untuk meminta petunjuk dalam menggunkan program.

Gambar 3.5 Menu Petunjuk

3.2 Persiapan Pembuatan *Watermark*ing

Pada pembuatan *watermark*ing ini terdapat dua tahapan yaitu tahap penyisipan dan tahap ekstraksi.

Pada tahap penyisipan, pertama dilakukan terhadap citra berdasarkan path yang dikirimkan oleh kelas MIDlet. Langkah selanjutnya adalah melakukan pembacaan terhadap byte dari citra tersebut. Nantinya byte ini akan disispkan dengan menggunakan metode LSB berdasarkan byet yang sudah dibaca sebelumnya. Setelah melakukan pembacaan byte, dilakukan pembacaan teksdan kunci yang juga dikirmkan oleh kelas MIDlet.

Proses selanjutnya adalah memroses teks yang akan disisipkan. Pada tahap ini dilakukan dua tahap inti yaitu pembacaan terhadap byte panjang teks dan pembacaan terhadap byte teks. Pembacaan byte panjang teks dilakukan dengan tujuan mendapatkan panjang teks yang harus dicari pada citra ketika nantinya dilakukan ekstraksi. Byte dari panjang teks ini akan ikut disisipkan di dalam citra digital. Proses pembacaan byte panjang teks ini nantinya akan dikonversi sehingga menghasilkan byte dengan jumlah sebanyak 4 byte. Hal ini dikarenakan nilai maksimal dari suatu integer adalah 2^{32}-1 yang setara dengan 32 bit. Dengan demikian seluruh bilangan integer dapat dikonversi menjadi 4 byte. Selanjutnya hasil byte tersebut akan dikonversi menjadi bilangan biner dan disimpan dalam suatu variabel. Sedangkan tahap selanjutnya adalah melakukan pembacaan terhadap byte dari teks itu sendiri. Pada tahap ini, setiap karakter dari teks akan disimpan dalam 1 byte. Byte tersebut kemudian akan dikonversi menjadi bilangan biner. Hasilnya kemudian disimpan di dalam variable yang menyimpan bilangan biner dari panjang teks tadi. Hasil dari proses ini adalah suatu variable yang menyimpan seluru bilangan biner dari panjang teks dan teks itu sendiri dan siap untuk disisipkan di LSB citra.

Langkah selanjutnya adalah melakukan penyisipan bilangan biner tadi ke dalam byte LSB citra. Pada tahap ini adalah dilakukan penggantian dari LSB citra menjadi bilangan biner. Hal ini dilakukan dari bilangan biner pertama hingga bilangan biner terakhir. Bilangan yang dihasilkan dari pembacaan citra digital khususnya bmp terdiri dari 4 bagian dan masing-masing bagian merupakan 1 byte. Adapun empat bagian tersebut adalah byte pertama menunjukkan bilangan alpha, yaitu skala yang menunjukkan kenampakkan suatu citra. Byte selanjutnya menunjukkan nilai R, yaitu besarnya kemerahan pada bit tersebut. Sedangkan dua byte sisanya sama seperti byte pertama dimana byte ini mewakili masing-masing warna hijau dan biru. Satu kesatuan ini sering disebut dengan satu satuan piksel.

Selanjutnya penggantian LSB citra dilakukan berdasarkan informasi kunci masukkan pengguna. Kunci yang dimasukkan pengguna akan membangkitkan bilangan random. Bilangan random ini nantinya akan menunjukkan byte ke berapa LSBnya akan digantikan dengan bilangan biner. Setelah selesai, maka proses *watermark*ing pun selesai.

Tahap kedua adalah tahap ekstraksi. Tahap ini hampir sama dengan teknik *watermark*ing, bedanya jika pada tahap *watermark*ing dilakukan penyisipan, sedangkan pada tahap ekstraksi dilakukan pembacaan. Jadi prosesnya adalah dilakukan pembacaan byte terhadap citra digital yang ingin dibaca. Pada aplikasi ini pembacaan citra digital dilakukan berdasarkan path yang dikirimkan dari kelas MIDlet. Setelah didapatkan byte dari citra tersebut kemudian dikonversi ke dalam bilangan biner. Setelah itu dilakukan proses pembacaan terhadap LSB dari byte tersebut. Banyaknya pembacaan bisa diketahui berdasarkan informasi panjang teks yang terletak pada 32 byte pertama pada citra digital. Jadi pertama harus dilakukan pembacaan terhadap LSB 32 byte pertama dari citra. Setelah didaptkan panjang teks, kemudian dilakukan pencarian teks pada byte citra tersebut. Kunci masukkan pengguna akan membangkitkan bilangan acak. Pembacaan dilakukan terhadap hasil dari bilangan acak tersebut. Jadi pembacaan LSB pada byte keberapa ditentukan oleh hasil bilangan acak yang dihasilkan. Setelah selesai dilakukan pembacaan, maka hasil kumpulan bit tersebut kemudian dijadikan satu dan dipecah menjadi masing-masing delapan. Setelah itu dilakukan konversi ke dalam bentuk karakter dan didapatkan teks yang disisipkan di dalam citra tersebut. Beberapa kendala dalam tahap ini adalah citra digital yang sudah ter-*watermark* tidak bisa dikembalikan seperti semula karena tidak dilakukan penyimpana terhadap LSB byte lama. Hal ini membuat *watermark* akan tetap terbawa pada citra.

3.3 Memperkuat *watermark*

Pada bagian ini saya tertarik untuk mengembangkan seikit metode LSB dari *watermark*ing. Pada makalah ini saya mengembangkan suatu permasalahan dimana terjadi penyalahgunaan pada suatu citra dan tidak bisa dibuktikan bahwa citra tersebut sudah dimanipulasi. Contohnya adalah suatu citra yang disisipi *watermark* kemudian dimanipulasi dan kebetulan *watermark* tersebut masih berada pada citra digital hasil manipulasi, tentu saja pemiliki asli citra tersebut tidak dapat menunjukkan bahwa citra tersebut sudah dimanipulasi. Untuk dapat menyelesaikan masalah tersebut, metode LSB perlu dikembangkan. Metodenya adalah pada tahap penyisipan, sebelum dilakukan penyisipan bilangan biner teks ke dalam byte citra, byte citra terlebih dahulu disisipkan oleh suatu teks yang berulang misalkan “sisip” hingga semua byte citra telah tersisispkan. Jadi ketika pada saat ini dilakukan ekstraksi, akan dihasilkan teks “sisipsisipsisip…”. Setelah dilakukan penyisipan ini, barulah dilakukan penyisipan bilangan biner dari teks masukkan pengguna.

Dengan menggunakan metode ini, tidak akan mengubah hasil dari teks yang disispkan. Jika dilakukan ekstraksi maka akan didapatkan teks yang disisipkan beserta informasi tentang hasil penyisipan dummy teks. Kerusakan pada dummy teks yang dihasilkan adalah maksimal sebanyak k karakter dimana k=n*8 dimana n adalah jumlah karakter pada teks masukkan pengguna. Meskipun demikian, metode ini tetap dapat digunakan di dalam *watermark*ing pada *mobile phone* dan dapat mengatasi permasalahan di atas karena ketika dilakukan manipulasi citra secara langsung, maka banyak kerusakan yang terjadi pada dummy teks lebih besar dari k karakter.

3.3 Perhitungan PSNR

Pada pengecekan PSNR ini dilakukan pengecekan terhadap beberapa citra sebagai perbandingan dari hasil PSNR. Perhitungan PSNR ini menggunakan perumusan PSNR

$$PSNR = 20 \times \log_{10}\left(\frac{256}{rms}\right)$$

$$rms = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_{ij} - \hat{I}_{ij})^2}$$

Setelah dilakukan implementasi, selurih *file* uji menghasilkan nilai di atas 30. Hal ini menunjukkan bahwa citra yang dihasilkan melalui metode *watermark* tergolong baik.

IV. KEKURANGAN DAN BATASAN

4.1 Pengaksesan *filesystem* java

Pada saat implementasi program pada *mobile phone*, platform java yang terdapat pada mobile tersebut tidak mengijikan dilakukannya pengaksesan terhadap *filesystem*-nya. Java yang mayortas menjadi platform suatu *mobile phone* cenderung melindungi perangkatnya dari pengaksesan yang tidak terpercaya. Java akan memproteksi sistemnya terutama *file system*-nya sehingga

tidak bisa dilakukan perubahan melalui suatu program yang belum dipercaya oleh Java. Pada program yang dibuat kali ini merupakan program yang tergolong asing oleh Java. Oleh karena itu sangat susah untuk mengakses direktori pada *mobile phone*. Berbeda dengan PC yang bisa dengan mudah dilakukan pengaksesan dir-nya melalui program biasa. Untuk mengantisipasi ini dilakukan permintaan ijin terhadap Java. Permintaan ijin ini dilakukan ketika melakukan pengaksesan *file* system pada java. Java akan meminta konfirmasi terhadap pengguna setiap kali pengguna melakukan suatu aksi yang mengakses dir *mobile phone*. Hal ini juga membuat beberapa fungsi *command* pada kelas MIDlet tidak berfungsi. Diharapkan kedepannya program ini dapat lebih dikembangkan lagi hingga bisa mengatasi permasalahan *entrust application* pada java.

Gambar 4.1 System proteksi java

4.2 Penanganan JPG

Program masih belum mampu menangani permasalahan JPG pada *mobile phone*. Pada saat dilakukan implementasi kamera, hasil ekstensi yang dihasilkan selalu berupa JPG. Satu-satunya cara agar masalah ini bisa diatasi adalah dengan membuat program dapat memproses *file* JPG.

Setelah dilakukan beberapa studi literature, hampir semua perangkat mobile menghasilkan gambar dengan format JPG dimana citra dengan format JPG merupakan hasil kompresi dan kualitasnya sendiri juga berbeda dibandingkan dengan format *file* lain misalnya (.bmp). Jika *file* dari (.bmp) dirubah menjadi JPG, maka kualitasnya akan menurun hingga 10 kali karena dilakukannya proses kompresi data.

VI. KESIMPULAN

*Watermark*ing bisa digunakan sebagai salah satu metode dalam menyelesaikan masalah penyalahgunaan citra digital dengan tingkat PSNR yang berada di atas 30db.

REFERENCES

[RIN05] Munir, Rinaldi. (2005). *Diktat Kuliah IF5054 Kriptografi*. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung.

[NO109] Noname
http://j2mesamples.blogspot.com/2009/02/file-connection-using-j2me-api-jsr-75.html
Waktu : Minggu, 18 Maret 2012 [6:00]

[ZAN10] Zanuar
http://zanuar.com/2010/01
Waktu : Minggu, 18 Maret 2012 [18:00]

[WIK10] Wikipedia
http://en.wikipedia.org/wiki/JPEG
Waktu : Minggu, 18 Maret 2012 [18:00]

[JAV10] Java
http://developers.sun.com/mobility/apis/articles
Waktu : Minggu, 18 Maret 2012 [18:00]

[ARD11] Ardiansyah
http://ardiansyahtkj86.wordpress.com/2011/03/12/sekilas-tentang-j2me
Waktu : Minggu, 18 Maret 2012 [18:00]

[ARD11] Bnet
http://stackoverflow.com/questions/5038680/file-connectionj2me
Waktu : Minggu, 18 Maret 2012 [18:00]

[WIK12] Wikipedia
http://en.wikipedia.org/wiki/BMP_file_format
Waktu : Minggu, 19 Maret 2012 [13:30]

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Maret 2012

I Nyoman Prama Pradnyana
13509032