

# Digital Image Watermarking Pada BlackBerry

Aris Feryanto

Program Studi Teknik Informatika ITB  
Jl Ganesha 10, Bandung  
email: aris\_feryanto@yahoo.com

## Abstract

Saat ini perkembangan penggunaan *BlackBerry* di Indonesia sangat pesat. Bahkan statistik menunjukkan bahwa jumlah permintaan perangkat *BlackBerry* termasuk dalam 5 besar jumlah permintaan mobile device di Indonesia. Dari keseluruhan pengguna *BlackBerry*, sebagian besar pernah menggunakan aplikasi facebook for *BlackBerry* untuk melakukan update status, melihat-lihat profil teman, dan upload foto yang diambil menggunakan kamera pada *BlackBerry*. Foto-foto yang di-upload ini bisa saja dengan mudah disalahgunakan oleh orang lain (di-crop, diubah, atau ditambahkan gambar).

Dalam makalah yang akan dibuat, penulis akan mencoba membangun aplikasi yang diberi nama iMark untuk mendukung implementasi digital watermarking sebagai solusi untuk melindungi kepemilikan citra yang diambil menggunakan kamera pada perangkat *BlackBerry*.

Dalam melakukan watermarking, akan disisipkan gambar watermark hitam putih berukuran kecil ke dalam citra yang diambil menggunakan kamera pada *BlackBerry*. Kemudian, iMark juga dapat melakukan ekstraksi watermark dari sebuah citra dan menampilkannya, sehingga dapat dideteksi oleh user, apakah citra tersebut adalah citra yang sah.

**Kata Kunci:** watermarking, BlackBerry, deteksi perubahan citra dari hasil ekstraksi

## 1. PENDAHULUAN

Perkembangan penggunaan perangkat *mobile* dewasa ini semakin berkembang dengan pesat. Hampir semua penduduk di kota-kota besar pasti memiliki perangkat *mobile*. Di Indonesia, salah satu perangkat *mobile* yang cukup populer saat ini adalah *BlackBerry*. Menurut statistik, permintaan akan perangkat *BlackBerry* di Indonesia menempati lima besar permintaan perangkat *mobile* secara nasional.

*Blackberry* terkenal karena banyak kemudahan yang bisa didapatkan dengan menggunakan *BlackBerry*, salah satunya adalah kemudahan dalam melakukan aktivitas *online* (*social networking*, *chatting*, dan *email*). Sangat banyak pengguna *BlackBerry* yang menggunakan aplikasi untuk aktivitas *online*, salah satunya adalah aplikasi *facebook for BlackBerry*. Dengan menggunakan aplikasi ini, pengguna dapat mengunggah foto yang ada pada perangkat *BlackBerry* atau foto yang diambil dengan kamera *built-in* ke jejaring sosial *facebook* secara langsung.

Karena kemudahan seperti ini, banyak sekali pengguna *BlackBerry* yang mengunggah foto dari perangkat *BlackBerry*. Namun, sebagian besar pengguna tidak mengetahui bahwa foto-foto yang telah diunggah dapat dengan mudah diunduh oleh orang lain, diubah (*crop*, *resize*, *add clip art*, atau menyisipkan gambar lain), kemudian diunggah lagi untuk dipublikasikan. Hal ini kemudian menimbulkan pertanyaan mengenai originalitas dari setiap gambar

yang beredar di internet. Bahkan saat ini, tidak sedikit kasus hak cipta mengenai hasil foto dari kamera diperdebatkan.

Salah satu solusi untuk masalah ini adalah dengan menerapkan *digital image watermarking* pada perangkat *BlackBerry* yang sering digunakan untuk mengambil dan mengunggah foto.

## 2. DIGITAL WATERMARKING

Secara umum, *digital watermarking* adalah sebuah proses menyisipkan informasi ke dalam sebuah sinyal digital dengan sebuah cara yang sulit untuk dihilangkan. Jika sebuah sinyal yang mengandung *watermark* disalin, maka informasi yang disisipkan juga akan dibawa dalam salinannya. Sebuah sinyal dapat membawa beberapa *watermark* yang berbeda pada suatu saat. Jika sinyal digital sebagai media penyisipan informasi adalah sebuah gambar, maka hal ini disebut *digital image watermarking*.

*Watermarking* sebenarnya sudah ada sejak akhir abad ke-13. Pada masa itu, pabrik kertas di Fabriano, Italia, sudah membuat kertas yang diberi *watermark* atau tanda-air dengan menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan, terbentuklah suatu kertas yang ber-*watermark*. Kertas ini biasanya digunakan oleh seniman atau sastrawan untuk menulis karya mereka. Kertas yang telah memiliki tanda-air tersebut sekaligus dijadikan identifikasi bahwa karya seni pada

kertas tersebut adalah milik mereka.

Ide *watermarking* pada data digital dikembangkan di Jepang pada tahun 1990 dan di *Swiss* pada tahun 1993. *Digital watermarking* semakin berkembang seiring dengan semakin meluasnya penggunaan internet, objek digital seperti video, citra, dan suara yang dapat dengan mudah digandakan dan disebarluaskan.

### 2.1. Kategori watermarking

*Digital watermarking* dapat dikategorikan dengan beberapa cara.

#### Media Penyisipan

Berdasarkan sinyal tempat penyisipan atau *host signal*, ada beberapa jenis watermarking, yaitu:

1. *Audio watermarking*,
2. *Image watermarking*,
3. *Video watermarking*, dan
4. *Text watermarking*

#### Visibility

*Watermarking* juga dapat dikategorikan sebagai:

1. *Invisible watermarking*  
Dalam *invisible watermarking*, informasi atau *watermark* ditambahkan sebagai data digital ke audio, gambar atau video, namun informasi tersebut tidak dapat ditangkap secara langsung (walaupun sebenarnya mungkin untuk mendeteksi bahwa ada informasi yang tersembunyi). *Watermark* dapat ditujukan untuk penggunaan yang luas dan dibuat mudah untuk mengekstraknya atau mungkin juga sebagai satu bentuk steganografi, dimana sebuah kelompok mengkomunikasikan sebuah pesan rahasia yang tersimpan di sinyal digital. Dalam kedua kasus ini, tujuan yang ingin dicapai adalah menyisipkan tanda kepemilikan atau informasi lainnya ke dalam sinyal dengan sebuah cara agar tanda atau informasi tersebut sulit untuk dihilangkan.
2. *Visible watermarking*.  
Berbeda dengan *invisible watermarking*, informasi yang ditambahkan dalam *visible watermarking* dapat ditangkap dengan mudah oleh indera manusia. Pada umumnya *visible watermarking* hanya digunakan sebagai tanda kepemilikan sebuah dokumen atau media digital.

#### Proses Verifikasi

Berdasarkan proses verifikasi, watermarking juga dapat dikategorikan sebagai:

1. *Blind watermarking*  
Pada *blind watermarking*, proses verifikasi *watermark* dapat dilakukan tanpa memerlukan media asal atau aslinya.
2. *Non-blind watermarking*  
Sedangkan pada *non-blind watermarking*, proses

verifikasi *watermark* memerlukan media asal.

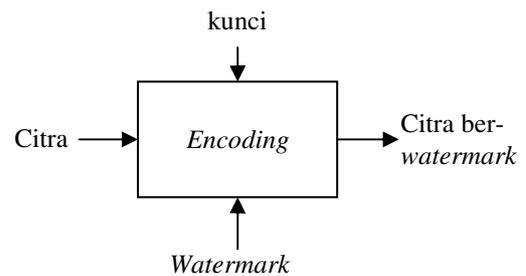
#### Robustness

1. *Fragile watermarking*  
Sebuah *watermark* dikatakan *fragile* apabila hanya dengan sedikit modifikasi pada media, *watermark* tidak dapat dideteksi kembali. *Watermark* kategori ini biasanya digunakan untuk *tamper proofing* atau identifikasi perubahan dari sebuah media digital.
2. *Semi-fragile watermarking*  
Sebuah *watermark* dikatakan *semi-fragile* bila dapat bertahan pada transformasi yang kecil atau sedikit, namun gagal dideteksi setelah mengalami transformasi yang rumit.
3. *Robust watermarking*  
Sebuah *watermark* dikatakan *robust* apabila dapat dipertahankan melalui serangkaian transformasi. *Watermark* ini biasanya digunakan dalam aplikasi *copy protection* untuk menyimpan informasi salinan dan kontrol akses.

Dari berbagai kategori *watermarking* ini, pembahasan aplikasi dalam makalah ini tergolong sebagai *invisible blind fragile image watermarking*.

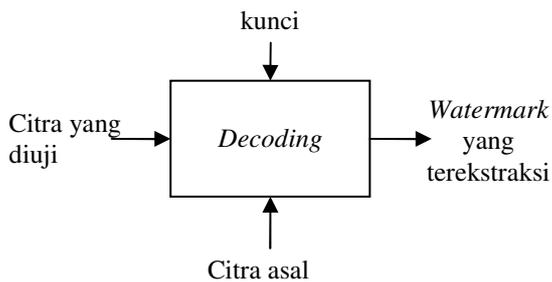
### 2.2. Penyisipan dan verifikasi watermark

Di sini hanya ditinjau mengenai *watermarking* pada citra digital. Proses penyisipan *watermark* ke dalam citra disebut *encoding* (Gambar 2.2.1). Sedangkan sub-proses ekstraksi dalam proses verifikasi disebut *decoding* (Gambar 2.2.2).



Gambar 2.2.1 Proses *encoding watermark* pada citra digital

Proses *encoding* dapat disertai dengan pemasukan kunci atau tidak memerlukan kunci. Kunci diperlukan agar *watermark* hanya dapat diekstraksi oleh pihak yang sah. Kunci juga dimaksudkan untuk mencegah *watermark* dihapus oleh pihak yang tidak berhak.

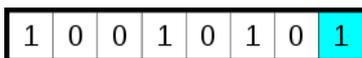


**Gambar 2.2.2** Proses *decoding watermark* pada citra digital

Proses *decoding* atau ekstraksi *watermark* merupakan sub-proses dalam melakukan verifikasi *watermark*. Tujuan dari *decoding* adalah untuk mengungkap *watermark* dari dalam citra. *Decoding* dapat dilakukan dengan mengikutsertakan citra asal atau tidak, tergantung dari jenis *decoding*-nya. Beberapa skema *watermarking* menggunakan citra asal dalam proses *decoding* untuk meningkatkan kualitas hasil ekstraksi.

Setelah didapatkan *watermark* yang terekstraksi, maka dilakukan proses perbandingan dengan *watermark asli* untuk mengambil keputusan apakah citra tersebut adalah citra yang sah.

Ada beberapa teknik algoritma untuk *image watermarking*. Salah satunya adalah dengan metode spasial. Metode ini menyisipkan *watermark* langsung pada nilai *byte* dari pixel pada citra. Tidak keseluruhan *byte* pada pixel citra diubah, hanya *least significant byte* (LSB) saja.



**Gambar 2.2.3** LSB (diberi *background*) dalam data 8 bit

Metode ini memanfaatkan properti dari LSB, yaitu ketika LSB diubah, tidak ada perubahan nilai yang signifikan pada *byte* tersebut. LSB yang digunakan tidak terbatas hanya pada satu bit, namun bisa merupakan kumpulan beberapa bit.

Dalam citra, perubahan beberapa bit dalam LSB hanya mengubah sedikit warna pada citra. Citra yang telah disisipi *watermark* pada pixel-pixelnya tidak akan banyak berubah dan mata manusia yang kurang peka dengan perubahan kecil warna pada gambar mengakibatkan manusia sukar membedakan mana gambar yang asli dan mana gambar yang sudah disisipi *watermark*.

### 2.3. Aplikasi *watermarking*

Sebuah aplikasi dari *watermarking* adalah dalam sistem perlindungan hak cipta, yang digunakan untuk menghindari penyalinan media digital secara ilegal. Dalam penggunaannya, alat penyalin mengambil *watermark* dari sinyal digital sebelum membuat sebuah salinan. Alat penyalin membuat sebuah

keputusan untuk menyalin atau tidak berdasarkan isi *watermark* tersebut. Contoh aplikasi *watermarking* yang lain adalah penelusuran sumber dari data digital. Sebuah *watermark* disisipkan ke dalam data digital pada tempat awal distribusi. Jika sebuah salinan nantinya ditemukan, *watermark* dari salinan tersebut bisa diambil dan sumber distribusinya dapat ditemukan. Teknik ini banyak digunakan untuk mendeteksi sumber distribusi dari film yang disalin secara ilegal. *Annotation* pada foto digital dengan deskripsi informasi adalah sebuah aplikasi lain dari *invisible watermarking*.

Beberapa contoh lain dari aplikasi *watermarking*:

- *Broadcast Monitoring* (berita televisi biasanya mengandung video yang sudah disisipi *watermark* dari agen internasional)
- *Covert Communication*

*Watermark* berbeda dengan *metadata* (informasi tambahan) pada media digital. Pada *digital watermarking*, informasi terkandung dalam media itu sendiri.

## 3. BlackBerry®

*BlackBerry* adalah nama serangkaian *smartphone* yang diproduksi oleh perusahaan Kanada, *Research In Motion* (RIM). Selain memiliki kemampuan sebagai *smartphone*, seperti memiliki aplikasi kalender, *todo-list*, *task manager*, dsb., *BlackBerry* juga terutama dikenal karena kemampuannya untuk mengirim dan menerima e-mail dimanapun saat bisa mendapatkan akses ke sebuah jaringan *wireless*.

Perangkat *BlackBerry* pertama kali diluncurkan pada tahun 1999 sebagai *pager* dua arah. Baru pada tahun 2002, *BlackBerry* seperti yang pada umumnya dikenal sekarang diluncurkan. Beberapa fitur utama *BlackBerry* adalah *push e-mail*, *mobile telephone*, *text messaging*, *internet faxing*, dan *web browsing*. *BlackBerry* pertama kali ditargetkan dengan fokus utama di bidang *e-mail*.

### 3.1. BlackBerry Operating System

RIM menyediakan sebuah *proprietary multi-tasking operating system* (OS) untuk *BlackBerry*. OS ini menyediakan dukungan untuk Java MIDP 1.0 dan WAP 1.2. Pengembang pihak ketiga dapat menulis *software* untuk *BlackBerry* dengan menggunakan API dari Java 2 Micro Edition (J2ME) seperti yang terdapat pada banyak perangkat ponsel, atau menggunakan API *proprietary* dari RIM. Jika menggunakan API dari RIM, ada beberapa fungsionalitas yang dibatasi dan harus di-*sign* secara digital untuk menggunakannya.

### 3.2. Pengembangan *software* pada BlackBerry

Pada dasarnya, ada dua jenis pengembangan utama

software pada BlackBerry, yaitu:

1. *Java development*

Karena BlackBerry Operating System dikembangkan dengan bahasa pemrograman Java, tentunya Java juga didukung dalam pengembangan software-nya. Pada dasarnya pengembangan berbasis Java di BlackBerry hampir sama dengan pengembangan aplikasi Java 2 Micro Edition (J2ME) pada perangkat ponsel pada umumnya, namun dilengkapi dengan API dari BlackBerry.

Pemrograman dalam lingkungan Java cukup powerful karena banyak fitur-fitur yang bisa diakses dengan menggunakan API yang disediakan. Perangkat utama yang disediakan untuk pengembangan di lingkungan Java adalah Java Development Environment atau biasa disingkat JDE. Selain itu, tersedia juga perangkat pengembangan sebagai plugin dalam editor pengembangan software lain yang sudah cukup terkenal seperti Eclipse.

2. *Web development*

Selain pengembangan di lingkungan Java, BlackBerry juga menyediakan pengembangan di lingkungan web. Pengembangannya juga menggunakan tool yang sudah terkenal untuk pengembangan web, seperti Eclipse dan Microsoft Visual Studio.

Pada implementasi image watermarking pada perangkat BlackBerry yang akan dijelaskan pada bab berikutnya, penulis menggunakan pengembangan di lingkungan Java. Alasan menggunakan lingkungan pengembangan Java adalah karena fitur-fitur seperti menangkap citra dari kamera BlackBerry atau mengolah file citra yang tersimpan, hanya tersedia pada lingkungan pengembangan Java.

**4. IMPLEMENTASI IMAGE WATERMARKING PADA BlackBerry®, iMark**

Penulis mengimplementasikan teknik digital image watermarking dengan metode spasial dalam sebuah aplikasi BlackBerry yang diberi nama iMark.

Pengguna dapat menggunakan iMark untuk mengambil foto dan secara otomatis menambahkan watermark sebelum foto disimpan. Selanjutnya, pengguna juga dapat menelusuri dan membuka file gambar dan mengekstrak watermark di dalamnya.



**Gambar 4.1 Screenshot aplikasi iMark yang berjalan di BlackBerry emulator**

Pada BlackBerry API, telah terdapat fungsi untuk mengambil citra dari kamera, serta encoding dan decoding citra dengan format JPEG. Perlu diketahui bahwa tidak disediakan encoding dan decoding untuk format citra yang lain untuk menangkap citra dari kamera. Berikut ini merupakan potongan kode untuk mengambil citra dari kamera pada BlackBerry.

```
import java.util.Vector;
import javax.microedition.media.Manager;
import javax.microedition.media.Player;
import javax.microedition.media.control.VideoControl;
import net.rim.device.api.ui.UiApplication;
...
/**
 * Screen untuk menangkap gambar dari kamera
 */
final class CameraScreen extends MainScreen
{
    private VideoControl _videoControl;
    private Field _videoField;
    private ObjectChoiceField _encodingField;
    private EncodingProperties[] _encodings;
    private ButtonField _photoButton;
```

```

/**
 * Constructor. Initializes the camera and creates the UI.
 */
CameraScreen()
{
    initializeCamera();
    initializeEncodingList();
    if(_videoField != null)
    {
        createUI();
    }
    else
    {
        add( new RichTextField( "Error connecting to camera." ) );
    }
}

/**
 * Initializes the Player, VideoControl and VideoField.
 */
private void initializeCamera()
{
    try
    {
        //Create a player for the Blackberry's camera.
        Player player = Manager.createPlayer( "capture://video" );

        //Set the player to the REALIZED state (see Player docs.)
        player.realize();

        //Grab the video control and set it to the current display.
        _videoControl = (VideoControl)player.getControl( "VideoControl" );

        if (_videoControl != null)
        {
            _videoField = (Field) _videoControl.initDisplayMode
(VideoControl.USE_GUI_PRIMITIVE, "net.rim.device.api.ui.Field");
            _videoControl.setVisible(true);
        }
        player.start();
    }
    catch(Exception e)
    {
        Dialog.alert( "ERROR " + e.getClass() + ": " + e.getMessage() );
    }
}

/**
 * Initialize the list of encodings.
 */
private void initializeEncodingList()
{...}

/**
 * Adds the VideoField and the "Take Photo" button to the screen.
 */
private void createUI()
{...}

/**
 * Create a screen used to display a snapshot.
 * @param raw A byte array representing an image.
 */
private void createImageScreen( byte[] raw )
{...}
}

```

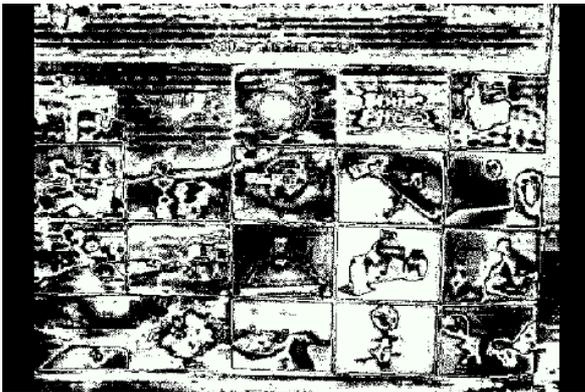
Setelah citra dari kamera berhasil diambil, maka akan dilakukan penambahan watermarking secara langsung ke citra tersebut. *Watermark* yang ditambahkan disini berupa gambar monokrom (hanya hitam dan putih) berukuran 20x20 pixel. Gambar *watermark* ini tidak hanya disisipkan satu kali pada ujung kiri atas citra,

namun di keseluruhan citra, sehingga diharapkan apabila dilakukan ekstraksi, *watermark* hasil ekstraksi adalah berupa pola *watermark* seukuran citra.



#### 4.1. Ujicoba watermarking

Berikut ini adalah hasil ujicoba watermarking pada BlackBerry. Hasil ujicoba ditampilkan dengan screenshot citra yang telah disisipi watermark dan hasil ekstraksi watermark dari gambar tersebut.

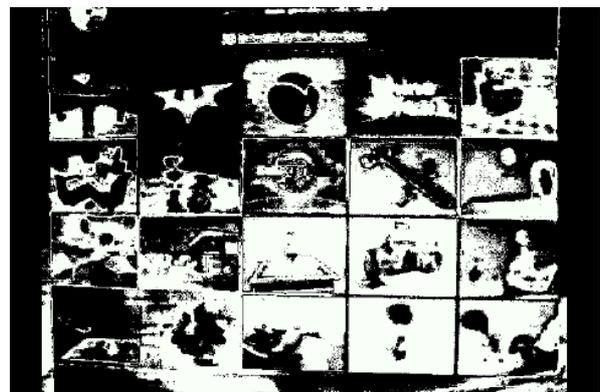


Gambar 4.1.1 Hasil penyisipan watermark dan ekstraksinya

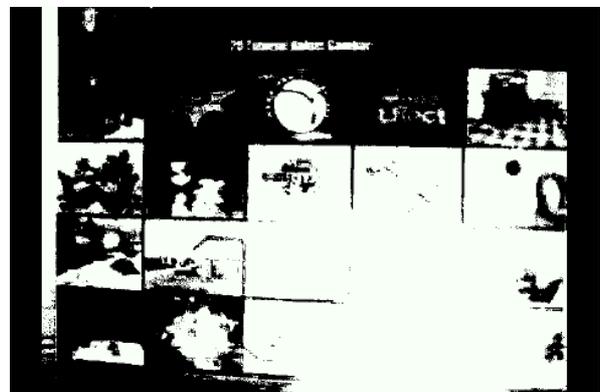
Dari hasil ujicoba dengan penyisipan 1 bit, hasil ekstraksi tidak sama seperti yang diinginkan. Watermark yang sudah ditambahkan terkesan seolah-olah hilang. Hal ini bisa terjadi karena format encoding yang digunakan adalah JPEG. Dalam encoding JPEG, citra akan ditransformasi, dikompres, diubah nilai kromanya, dsb., sehingga 1 bit yang ditambahkan mungkin saja hilang setelah melalui tahapan encoding JPEG tersebut.

Untuk itu, penulis mencoba menyisipkan watermark tidak hanya ke 1 bit pada LSB, namun beberapa bit. Hal ini memang memiliki efek yang buruk, yaitu pengaruh perubahan nilai komponen warna biru pada setiap pixel citra akan terlihat lebih jelas (bisa dibedakan oleh mata manusia). Hasil yang signifikan terlihat setelah penyisipan pada 4 bit LSB.

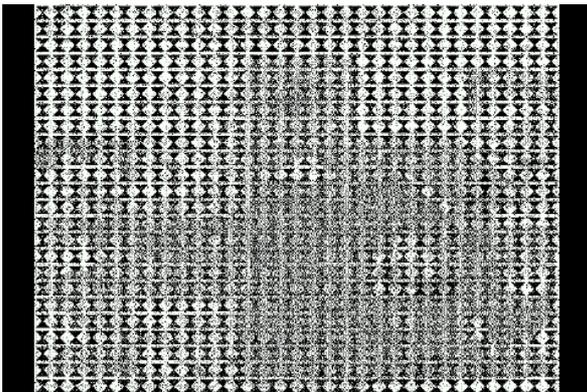
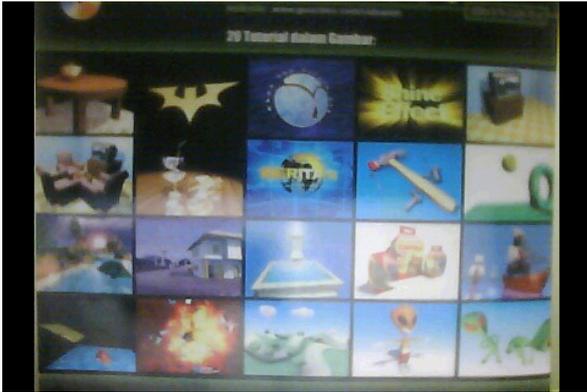
Berikut ini adalah pasangan citra hasil penambahan watermark dan hasil ekstraksinya.



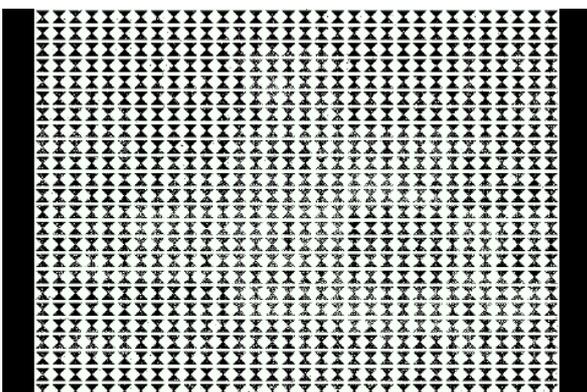
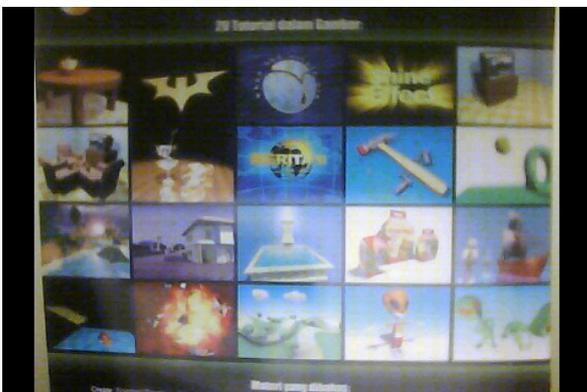
Gambar 4.1.2 Hasil penyisipan watermark (LSB = 2bit) dan ekstraksinya



Gambar 4.1.3 Hasil penyisipan watermark (LSB = 3bit) dan ekstraksinya



Gambar 4.1.4 Hasil penyisipan *watermark* (LSB = 4bit) dan ekstraksinya

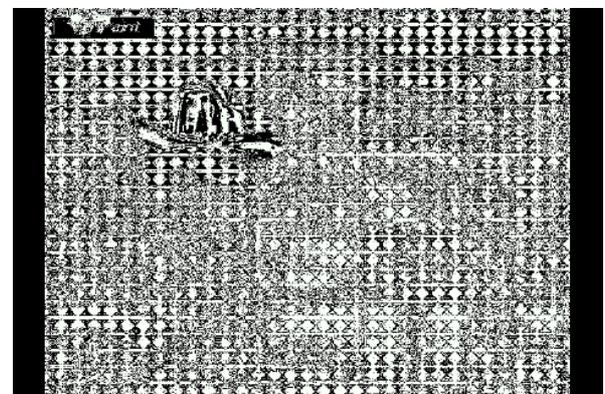


Gambar 4.1.5 Hasil penyisipan *watermark* (LSB = 5bit) dan ekstraksinya

Hasil ekstraksi yang paling bagus adalah ekstraksi *watermark* dengan LSB 5 bit. Namun, bila diperhatikan, citra hasil penambahan *watermark* terlihat memiliki cetakan pola *watermark* berwarna biru. Berbeda dengan penyisipan 1 bit hingga 4 bit, dimana citra hasil *watermarking* tidak terlalu terlihat pola *watermark*-nya.

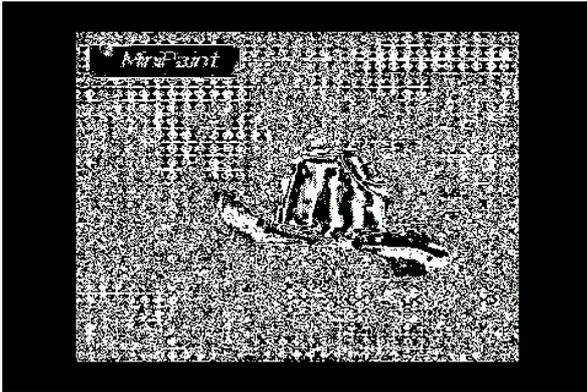
#### 4.2. Ujicoba serangan terhadap *watermarking*

Ujicoba berikutnya adalah apabila dilakukan serangan terhadap citra yang sudah di-*watermarking*. Dalam ujicoba ini akan dilakukan serangan dalam bentuk *resize* dan penambahan gambar lain. (Dalam proses *watermarking* dan ekstraksinya digunakan LSB 5 bit)



Gambar 4.2.1 Citra yang diberi tambahan gambar lain dan hasil ekstraksinya





**Gambar 4.2.2** Citra yang diresize, kemudian diberi tambahan gambar lain dan hasil ekstraksinya

Dari hasil ujicoba, serangan mengakibatkan hasil ekstraksi *watermark* pada citra menjadi hancur, terutama pada bagian yang ditambahkan gambar lain.

#### 4.3 Kesimpulan hasil ujicoba

1. *Watermarking* yang diimplementasikan di sini tergolong jenis *invisible blind fragile image watermarking*.
2. *Watermarking* pada JPEG tidak terlalu bagus untuk penyisipan pada LSB dengan jumlah bit sedikit.
3. Serangan terhadap citra yang telah di-*watermark* dapat dideteksi dengan melihat hasil ekstraksinya.
4. Implementasi yang sudah dibuat penulis dapat dikembangkan lagi dengan teknik *watermarking* yang lain, maupun ditambahkan kunci untuk meningkatkan keamanan.

## 5. KESIMPULAN

Kesimpulan yang dapat diambil dari hasil implementasi *watermarking* pada perangkat *BlackBerry* ini adalah:

1. *Digital image watermarking* sangat potensial untuk diterapkan pada perangkat *BlackBerry*.
2. Teknik spasial dapat diterapkan untuk *watermarking* pada *BlackBerry*, namun karena format gambar yang didukung hanya JPEG, diperlukan sedikit modifikasi agar *watermark* tetap dapat diekstraksi.
3. *iMark* dapat digunakan untuk *watermarking* dan ekstraksi *watermark* dari sebuah citra.
4. Citra hasil *watermarking* dengan *iMark* dapat diupload tanpa perlu mengkhawatirkan citra tersebut diubah oleh orang lain, karena bisa dideteksi mana citra yang asli dengan mengekstraksi *watermark*.

## DAFTAR REFERENSI

- [1] Mateev, Nikolay and Zhou, Lidong, *Watermarking JPEG Images*, <http://www.cs.cornell.edu/home/mateev/Watermark/watermarkingJPEG.html>
- [2] Munir, Rinaldi. (2005). *Diktat Kuliah IF5054 Kriptografi*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- [3] *BlackBerry developer*, <http://www.BlackBerry.com/developers/>