DIGITAL *IMAGE WATERMAKING* PADA *MOBILE DEVICE*

Budiono – NIM : 13503013

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl Ganesha 10, Bandung
E-mail : if13013@students.if.it.ac.id

Abstrak

Makalah ini membahas mengenai studi dan implementasi digital *image watermarking* dari sebuah citra pada *mobile device* yang mendukung fitur kamera. Digital *image watermarking* merupakan salah satu jenis *watermarking* untuk melindungi kepemilikan (*copyright*) dari sebuah citra. Sedangkan *Watermarking* merupakan bagian dari ilmu kriptografi dengan metode menyisipkan sebuah informasi ke dalam media tertentu. Implementasi digital *image watermarking* dalam makalah ini meliputi penerapan penggunaannya pada *mobile device* secara otomatis ketika citra diambil dari kamera *handphone*.

Sebuah perangkat lunak yang bernama JepretRight akan dibangun pada sebuah *mobile device* untuk mendukung implementasi dari digital *image watremarking*. Perangkat lunak ini dikembangkan dengan menggunakan bahasa pemograman JAVA khusunya J2ME yang menyediakan berbagai *interface* untuk melakukan pengambilan citra melalui kamera *mobile device*.

Perangkat lunak JepretRight tersebut kemudian akan digunakan untuk melakukan penyimpanan kunci yang digunakan untuk melakukan digital *image watermarking* sehingga tingkat keamanan dari citra yang dihasilkan dapat dijaga. Tingkat keamanan dari digital *image watermarking* ini dilakukan dengan mengoperasikan *pixels* dari citra yang dihasilkan serta tetap berusaha mempertahankan kualitas dari citra yang sebenarnya. Kemudian, perangkat lunak ini membandingkan citra yang asli dengan citra yang dihasilkan dari deskripsi citra yang ber*watermark*. Hasil pengujian menunjukkan bahwa *mobile device* memiliki kemampuan untuk melakukan proses digital *image wateramarking* pada sebuah citra yang diambil dari fitur kameranya tanpa menurunkan kualitas dari citra tersebut. Selain itu, implementasi dari digital *image watermarking* pada *mobile device* ini memiliki keterbatasan dalam melakukan proses algoritma penyisipan *watermark* dan ekstrasi/deteksi *watermark* dari citra yang dihasikannya dikarenakan keterbatasan melakukan komputasi dan kapasitas memori dari *mobile device* tersebut.

Kata Kunci : *Digital image watermarking*, citra, *copyright*, JAVA, J2ME, *moble device*, *pixel*, ekstraksi/deteksi, JepretRight.

1. Pendahuluan

Saat ini kebutuhan akan teknologi *mobile* semakin meningkat di kalangan masyarakat. Salah satunya adalah penggunaan fitur kamera yang terdapat dalam perangkat *mobile*. Dengan semakin banyaknya telepon seluler yang mendukung teknologi 3G, foto-foto yang dihasilkan dari perangkat *mobile* dapat dengan mudah di*upload* secara langsung ke internet dan dikonsumsi oleh masyarakat secara luas. Hal ini menimbulkan pertanyaan akan originilitas dari setiap gambar yang ada di internet. Bahkan tidak sedikit saat ini kasus-kasus hak cipta mengenai hasil foto kamera diperdebatkan. Akhirnya, muncul keraguraguan dari pihak pemilik perangkat *mobile* untuk meng*upload* foto mereka secara *online* di internet.

Untuk itu, perlu adanya suatu teknologi yang dapat melindungi hak cipta dari setiap foto yang dihasilkan dari perangkat *mobile* ini. Salah satu solusinya adalah dengan menerapkan *Digital Image Watermarking* pada perangkat *mobile* seperti *handphone* ataupun PDA (Personal Digital Assistant).

Saat ini digital *watermarking* muncul sebagai teknologi yang menjanjikan untuk melindungi hak atas kepemilikan [Zhao et al. 1998, Memon & Wong 1998]. Untuk melindungi hak cipta dari suatu hasil karya seperti foto maupun musik dapat dilakukan digital *watermarking* pada teks, video maupun audio dengan

menyisipkan informasi seperti informasi pemilik, informasi tujuan, ataupun informasi keaslian. Keefektifan dari digital *watermarking* tergantung kepada ketepatan ekstrasi dan enskripsi dari data yang ada dan ingin dilindungi keoriginalitasnya.

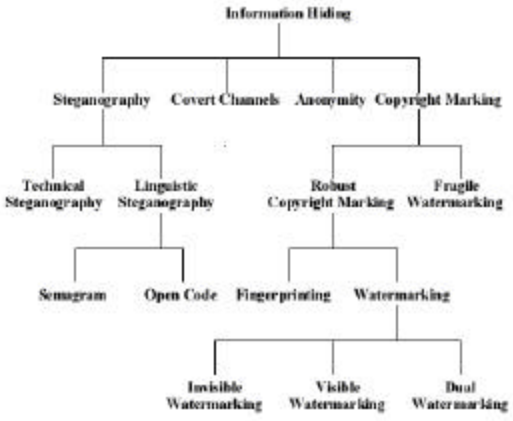
Mobile device seperti telepon seluler maupun PDA (Personal Digital Asistants) mengalami peningkatan penggunaan dalam mengubah, memunculkan, maupun mendengarkan konten dari data digital. *Mobile device* memiliki kemampuan komputasi yang terbatas yang dikarenakan oleh keterbatasan memori dan prosesor yang dimiliki [1].

2. Sejarah Penyembunyian Informasi [1]

Ide komunikasi secara tersembunyi telah ada selama kegiatan komunikasi itu sendiri. Kiasan tertua untuk menulis secara tersembunyi terdapat pada West yang muncul di Homer's Ilaid [9]. Metoda *Steganography* membuat rekor awalnya pada beberapa abad kemudian dalam beberapa cerita yang dibuat oleh Herodotus, bapak sejarah (the father of history) [10]. Beberapa darinya dapat juga ditemukan pada [7,19,23]. Kautilya's Arthasa'stra dan LalitaVista'ra, dan Vatsa'yana's Ka'masu'tra adalah bacaan Indian yang juga merupakan sedikit dari contoh terkenal penggunaan penulisan rahasia/*steganography.*

Beberapa contoh lainnya dari *steganography* dapat ditemukan pada [7,19,23]. Sebuah teknik penting adalah penggunaan *sympathetic inks.* Ovid dalam "Art of Love"-nya menyarankan penggunaan susu untuk menulis secara rahasia di mana tulisannya tidak dapat dilihat. Kemudian, dikembangkanlah *sympathetic inks* yang secara kimia dibuat-buat. *Sympathetic inks* tersebut digunakan pada masa perang dunia I dan II. Asal dari *Steganography* adalah sesuatu yang berhubungan dengan ilmu hayati dan fisiologis. Kata *Steganography* itu sendiri digunakan tahun 1500-an setelah munculnya buku Trithemius dengan judul "Steganographia". Secara keseluruhan, cabang dari *Steganography*, "linguistic *steganography*", mengandung bentuk bahasa dari penulisan tersembunyi. Terdapat beberapa "semagrams" dan "kode terbuka (open code)" [16, 19, 23]. Semagran adalah pesan rahasia yang tidak dalam bentuk tulisan. Sebagai contoh, sebuah sistem dapat menggunakan helaian rumput yang panjang dalam sebuah gambar sebagai tanda garis dalam kode morse dan helaian rumput yang pendek sebagai tanda titik. Catatan not musik pun dapat digunakan sebagai surat, tetapi hal ini akan membuatnya tidak seperti sebuah musik. *Open code* menggunakan ilusi atau kata-kata sandi. Sebagai contoh, pada perang dunia I, mata-mata Jerman menggunakan perintah tipuan dengan menggunakan cerutu sebagai representasi berbagai jenis kapal perang dan penghancur Inggris. Maka jika 5000 cerutu diperlukan di Portsmouth, hal itu berarti 5 kapal ada di Portsmouth.

Teknik *watermark*ing telah berkembang dari *steganography*. Penggunaan *watermark*s itu hampir berumur sama dengan industri kertas [32]. Nenek moyang kita telah menuangkan setengah *slurry* serat dan air ke lubang cetakan untuk mengumpulkan serat dan kemudian menyebarkan *slurry* ke dalam kerangka *deckle* untuk menambahkan bentuk dan keseragamannya, dan terakhir digunakan tekanan yang tinggi untuk mengeluarkan airnya dan melekatkan seratnya.



Gambar 1. Teknik Penyembunyian Informasi

Proses ini tidak mengalami banyak perubahan selama 2000 tahun. Hasil dari proses ini adalah *watermark* teknik penggambaran sebuah bentuk gambar di atas kertas, atau teks yang diperoleh dari cetakan negatif, selayaknya serat kertas yang ditekan dan dikeringkan. Kertas *watermark* telah digunakan secara luas sejak pertengahan abad. Penggunaan paling awal adalah untuk menyimpan merk dagang pabrik dalam sebuah produk sehingga keasliannya dapat ditetapkan dengan jelas tanpa menurunkan keindahan dan kegunaan produk. Dalam beberapa lama, *watermark* telah digunakan untuk menandakan komposisi dari kertas, termasuk kealamian serat yang digunakan. Negara-negara paling maju saat ini juga melakukan *watermark* pada kertas, mata

uang, dan materai untuk mempersulit tindakan pemalsuan.

Digitalisasi zaman saat ini telah membuat konsep *watermarking* berkembang mencakup impresi secara digital untuk membuktikan kepemilikan dan melindungi kepentingan pemilik. Namun, secara prinsip *watermark*s digital sama seperti kertas nenek moyang mereka. *Watermark*s digital menandai suatu tanda dari dokumen atau file yang diwarisinya. Apakah produk dari penekanan kertas atau pun transformasi kosinus tersendiri, *watermark* dari berbagai sudut pandang ditambahkan pada penyajian media sebagai jaminan keaslian, kualitas kepemilikan dan sumber.

3. *Watermarking*

Digital watermarking atau *watermarking* adalah teknik untuk menyisipkan informasi tertentu ke dalam data digital yang disebut *watermark* (tanda air). *Watermark* dapat berupa teks seperti informasi *copyright*, gambar berupa logo, data audio, atau rangkaian bit yang tidak bermakna. Penyisipan *watermark* dilakukan sedemikian sehingga *watermark* tidak merusak data digital yang dilindungi. Selain itu *watermark* yang telah disisipkan tidak dapat dipersepsi oleh indra manusia, tetapi dapat dideteksi oleh komputer dengan menggunakan kunci yang benar. W*atermark* yang telah disisipkan tidak dapat dihapus dari dalam data digital sehingga jika data digital tersebut disebar dan diduplikasi maka otomatis *watermark* di dalamnya akan ikut terbawa. *Watermark* di dalam data digital harus dapat diekstraksi kembali. *Watermarking* berguna untuk membuktikan kepemilikan, *copyright protection*, *authentication*, *fingerprinting*, *tamper profing*, dan *distribution tracing*.

Sejarah *watermarking* sudah dimulai sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi *watermark* atau tanda air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan, terbentuklah suatu kertas yang ber- *watermark*. Kertasini biasanya digunakan oleh seniman dan sastrawan untuk menulis karya mereka. Kertas yang sudah dibubuhi *watermark* tersebut sekaligus dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka.

Ide watermarking pada data digital dikembangkan di Jepang tahun 1990 dan di Swiss 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.

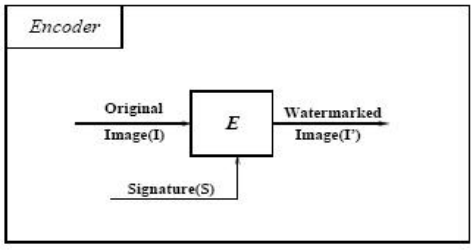
Saat ini kebanyakan data dan informasi disajikan dalam bentuk format digital, baik berupa teks, citra, audio, maupun video. Produk digital lainnya, mempunyai beberapa karakteristik, antara lain penggandaan terhadap data digital juga mudah dilakukan dan hasilnya tepat sama dengan aslinya, mudah didistribusikan melalui *magnetic disk* maupun internet, dan perubahan sedikit pada citra tidak mudah dipersepsi oleh indera penglihatan.

Digital *watermarking* telah dijadikan sebuah teknik untuk melindungi hak intelektual pada dunia digital. Pelindungan ini menyangkup pencegahan pengcopyan data digital secara ilegal dan memvalidasi data digital baik data digital audio, tekstual, visual, dan video. *Watermarking* dapat juga digunakan untuk membuktikan kepemilikan, mendeteksi keaslian dari *file* digital, menyisipkan informasi di dalamnya maupun menyembunyikan data.

Dalam melakukan *watermarking* secara umum, terdapat tiga bagian utama. Bagian tersebut diantaranya [2]:

- *Watermark*.
  Informasi yang akan digunakan untuk disisipkan pada data digital.

- *Encoder* (Algoritma y ang digunakan)



Gambar 2. Proses *Encoder*

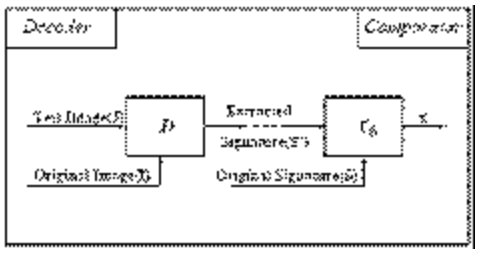
  E adalah fungsi *encoder* yang menerima masukan sebuah citra I dan tanda (informasi) S, dan menghasilkan citra baru yang disebut sebagai *watermarked image* I'. Secara matematik dapat dirumuskan sebagai :

$$E(I, S) = I' \qquad (1)$$

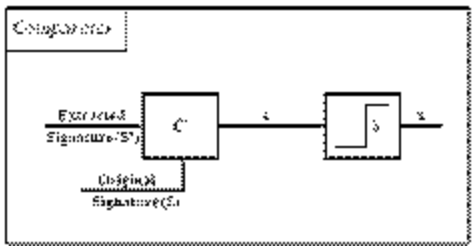
- *Decoder* dan *Comparator* (verifikasi atau ekstrasi atau deteksi algoritma)

$$D (J , I) = S' \qquad (2)$$

Fungsi *decoder* D menempatkan *image* J dan I sebagai parameter untuk menghasilkan sebuah informasi atau kunci yang ada dalam *image* yang ber*watermark* J.



Gambar 3. Fungsi *Decoder*



Gambar 4. Fungsi Comparator

Sebuah *watermarking* harus dapat dideteksi dan diekstrak untuk digunakan kembali.

Sebuah teknik *watermarking* yang handal harus memenuhi persyaratan berikut:

1. *Imperceptibility*: keberadaan *watermark* tidak dapat dipersepsi oleh indra visual. Hal ini bertujuan untuk menghindari gangguan pengamatan visual.
2. *Key uniqueness*: kunci yang berbeda seharusnya menghasilkan *watermark* yang berbeda. Ini berarti penggunaan kunci yang salah dapat menyebabkan hasil ekstraksi atau deteksi *watermark* yang salah pula.
3. *Noninvertibility*: secara komputasi sangat sukar menemukan *watermark* jika diketahui hanya data ber-*watermark* saja.
4. *Robustness*: *watermark* seharusnya tetap kokoh terhadap serangan yang dilakukan pada data ber- *watermark*. Ini berarti manipulasi yang dilakukan terhadap data ber- *watermark* tidak merusak *watermark* sehingga *watermark* masih dapat dideteksi.

Sebagian besar penelitian, publikasi, dan aplikasi di bidang *watermarking* ditujukan untuk citra digital, akan tetapi *watermarking* juga dapat diterapkan pada jenis data multimedia lain, seperti audio, video, dan teks. *Watermarking* pada video digital memerlukan teknik tertentu sehingga peralihan gambar dari satu *frame* ke *frame* lainnya harus tetap baik dan tidak terlihat dimodifikasi. *Watermarking* pada video digital memerlukan proses penyisipan yang lebih banyak, hal ini disebabkan ukuran file video digital yang relatif lebih besar daripada citra. Sedangkan untuk *watermarking* pada data audio, perlu ketelitian pada perancangan algoritma *watermarking* karena suara lebih sensitif daripada citra. Hal ini berarti suara digital lebih mudah rusak jika ditambahkan *watermark*. *Watermarking* pada dokumen teks menggunakan metode yang berbeda yaitu dengan cara menyisipkan spasi antara dua buah kata atau antara dua buah kalimat dalam suatu dokumen teks.

4. Pemodelan *watermarking*

Andaikan seorang bernama Alice mempunyai relasi R yang mengandung *n record*, dan Alice telah menandai sejumlah *w* record maka properti berikut harus dimiliki oleh pemodelan *watermarking*:

1. Detectability
   Alice harus bisa mendeteksi *watermark* miliknya dengan memeriksa *w* record dari suatu basis data. Jika pola bit (*watermark*) ternyata terdapat pada semua *w* record tersebut maka basis data tersebut adalah miliknya.
2. Robustness
   Watermark sebaiknya handal (*robust*) terhadap serangan untuk menghapus *watermark* tersebut.
3. Incremental Updatability
   Jika Alice memiliki relasi R yang sudah diberi *watermark* maka ia harus dapat melakukan *update* R tanpa menghilangkan *watermark*-nya. Selain itu, jika Alice melakukan penambahan atau penghapusan record maka *watermark* tersebut juga harus dapat di-*update*.

4. Imperceptibility
   Modifikasi pada basis data yang disebabkan oleh *watermarking* tidak boleh mempengaruhi kegunaan dari basis data tersebut. Selain itu, penggunaan perhitungan statistik seperti mean dan variansi dari atribut numerik tidak boleh terpengaruh secara signifikan.
5. Blind System
   Pendeteksian *watermark* sebaiknya tidak memerlukan informasi mengenai basis

data asli maupun *watermark* itu sendiri. Properti ini sangat penting karena memungkinkan *watermark* dapat dideteksi pada salinan dari relasi suatu basis data.

6. Key-Based System
Watermarking memiliki asumsi bahwa metode yang digunakan untuk menyisipkan *watermark* bersifat publik. Pertahanan terhadap *watermark* terletak pada pemilihan kunci privat.

5. Jenis Digital *Watermarking*

5.1 Visible Watermark

Watermarking jenis ini merupakan *watermarking* yang memiliki tujuan untuk meningkatkan perlindungan akan hak cipta. Selain itu, *watermarking* jenis ini juga digunakan untuk mengidentifikasi kepemilikan dari sebuah karya (originalitas).

5.2 Invisible Robust Watermark

Watermarking jenis ini untuk mendeteksi ketidaktepatan dari sebuah citra. Selain itu, jenis ini biasanya digunakan untuk menerangkan kemepilikan.

5.3 Invisible Fragile Watermark

Watermarking jenis ini digunakan oleh perangkat kamera yang cukup handal. Dan proses *watermarking* dilakukan ketika pengambilan gambar.

Berdasarkan berbagai jenis dari *watermarking* tersebut, makalah ini akan sesuai dengan jenis *watermarking Invisible Fragile Watermark*.

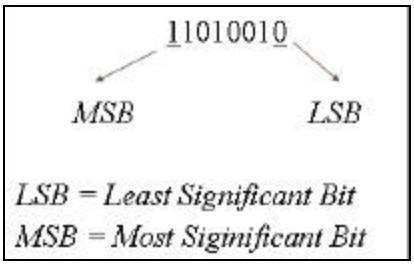
6. Karakteristik *Invisible Fragiles Watermarking* yang Baik[2].

- *Invisible Watermarking* harusnya tidak dapat diketahui oleh *viewer* dan tidak menurunkan kualitas dari konten yang sebenarnya.
- *Invisible Fragiles Watermarking* harusnya dapat dimodifikasi ketika ada suatu niai *pixel* gambar yang diubah.
- Untuk gambar yang berkualitas, jumlah *pixel* yang dimodifikasi harus sekecil mungkin.

7. Image *Watermarking*

Jumlah dari teknik algoritma untuk *image watermarking* sangat sedikit. Salah satunya adalah metode spasial[3]. Metode ini menyisipkan *watermark* langsung pada nilai *byte* dari pixel citra.

Metode ini dapat dilakukan dengan melakukan pergantian bit LSB dengan bit data.



Gambar 5. LSB dan MSB

Mengubah bit LSB hanya mengubah nilai byte satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya.

Dikarenakan implementasi *image watermarking* pada perangkat *mobile* memiliki keterbatasan melakukan komputasi, dalam makalah ini akan menerapkan metode spasial. Selain metode spasial, terdapat juga metode transformasi.

8. Serangan pada *Watermark*s [1]

Citra ber-*watermark* biasanya diselewengkan untuk kepentingan tertentu, beberapa penyelewengan yang dilakukan secara sengaja adalah kompresi dan transmisi bunyi dan hal-hal seperti pemotongan (*cropping*), *filtering*, dan lain-lain.

- Kompresi Lossy : Banyak skema kompresi seperti JPEG dan MPEG yang kemungkinan besar dapat menurunkan kualitas data melalui kehilangan sejumlah data yang tidak dapat dikembalikan
- Distorsi Geometric : Distorsi *Geometric* lebih spesifk pada citra video termasuk beberapa operasinya antara lain memutar (*rotation*), *translation, scalling,* dan pemotongan (*cropping*).
- Operasi Pemrosesan Sinyal secara Umum (*Common Signal Processing Operations*) : termasuk di dalamnya hal-hal berikut :

1. Konversi D/A
2. Konversi A/D
3. Penarikan contoh kembali (resampling)
4. Pengukuran kembali (*Requantization*)
5. *Dithering distortion*
6. Linear filtering seperti high pass dan low pass filtering
7. Non-linear filtering seperti median filtering
8. Reduksi warna
9. Konstanta penyeimbang tambahan pada nilai *pixel*
10. Tambahan kegaduhan Gaussian dan Non gaussian
11. Pertukaran *pixel* secara lokal

Serangan lain yang disengaja :

1. Pencetakan dan pemindahan kembali

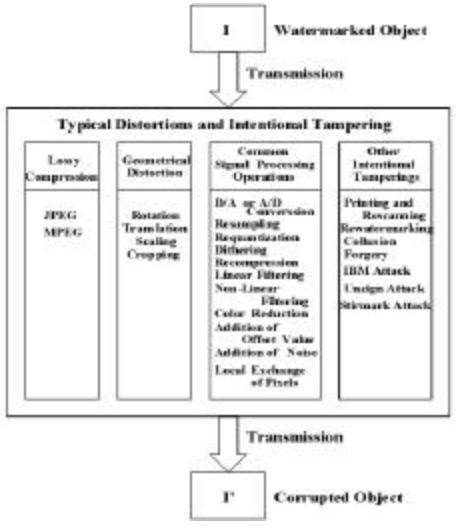
2. *Watermarking* citra yang sudah ber-*watermark* (*watermarking* kembali)

3. Kolusi : sejumlah penerima sah dari citra seharusnya tidak dapat meng-*collude* salinan ber-*watermark* yang berbeda untuk membangkitkan salinan yang tidak ber-*watermark* dari sebuah citra (dengan meratakan semua citra ber-watermark)

4. Pemalsuan *(Forgery*) : sejumlah penerima sah dari citra seharusnya tidak dapat melakukan collude untuk membentuk salinan dari citra ber-watermark dengan memasukkan watermark yang sah dari seseorang yang tidak berada pada kelompok dengan maksud penyusunan kelompok ketiga.

5. Penyerangan IBM [155,157] : seharusnya tidak mungkin untuk menghasilkan produk -asli tetapi palsu- yang memiliki performa layaknya produk yang asli. Begitu juga dalam hasil ekstraksi produk ber-*watermark* yang dinyatakan asli oleh pemilik produk -asli tetapi palsu- tersebut.

6. Unzign dan Stirmark telah menunjukkan kesuksesan luar biasa dalam menghilangkan data yang tersimpan/melekat pada program komersial yang ada



Gambar 6. Serangan pada *watermark*

8. JAVA

Bahasa pemrograman Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (Star Seven).

Setelah era Star Seven selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.

Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an,

Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemeritaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "bapak java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling.

Java adalah teknologi dan bahasa pemrograman yang berjalan pada multiflatform sesuai dengan semboyannya yaitu "*Write Once, Run Anywhere*". Pada site official Java dari Sun yaitu http://java.sun.com bisa ditemui tiga pembagian paket Java yaitu :

- Java 2 Enterprise Edition (J2EE).
- Java 2 Standart Editon (J2SE).
- Java 2 Micro Edition (J2ME).

Penjelasan paling simple atas pembagian tersebut berdasarkan atas perangkat keras yang digunakan.

a. Paket J2EE digunakan pada perangkat keras yang mempunyai spesifikasi dan memory yang besar seperti pada komputer server.
b. Paket J2SE digunakan pada perangkat keras seperti komputer desktop.
c. Paket J2ME digunakan pada perangkat yang memiliki memory kecil seperti ponsel, pager atau PDA.

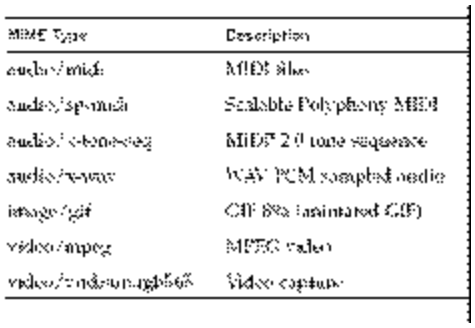
Paparan singkat di atas adalah penjelasan singkat mengenai Java dan sedikit gambaran dimana paket J2ME digunakan. Sebenarnya masih panjang penjelasan tentang Java dan paket J2ME, tapi tidak dibahas pada tulisan ini, mungkin bisa menjadi pekerjaan rumah buat para pembaca yang tertarik akan Java atau J2ME.

9. Implementasi *Image Watermarking* dengan J2ME

J2ME merupakan teknologi yang dikembangkan oleh Sun untuk aplikasi yang berjalan diatas sebuah perangkat *mobile* dalam hal ini *handphone*.

Di dalam J2ME, untuk mengambil sebuah citra / foto melalui kamera *hanphone* dapat dilakukan dengan menggunakan *Mobile Media* API.

Mobile Media API merupakan API dari Java yang mengkhususkan sebuiah *mobile device* untuk dapat menjalankan dan mengakses fitur multimedia yang terdapat pada *mobile device* tersebut.



Gambar 7. *Mobile Media* API

Setiap citra / foto yang dihasilkan dari kamera *handphone* oleh MMAPI disimpan dalam sebuah *Array of pixels* yang berisi kumpulan warna pada setiap citra yang dihasilkan. *Array of pixels* ini nantinya akan digunakan sebagai media untuk melakukan *image watermarking* pada *mobile device.*

Berikut hasil implementasi program dalam bahasa pemograman Java untuk menangkap citra dari kamera *handphone* :

```
import java.io.IOException;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;
import javax.microedition.midlet.MIDlet;
import javax.microedition.media.control.VideoControl;

public class JepretRight extends MIDlet implements CommandListener {

    private Display display;
    private Form form;
    private Command exit,back,capture,camera;
    private Player player;
    private VideoControl videoControl;
    private Video video;

    public JepretRight() {

        exit = new Command("Exit", Command.EXIT, 0);
        camera = new Command("Camera", Command.SCREEN, 0);
        back = new Command("Back", Command.BACK, 0);
        capture = new Command("Capture", Command.SCREEN, 0);

        form = new Form("Capture Image");
        form.addCommand(camera);
        form.setCommandListener(this);
    }

    public void showCamera() {
        try {
            player = Manager.createPlayer("capture://image");
            player.realize();

            videoControl = (VideoControl)player.getControl("VideoControl");
            Canvas canvas = new VideoCanvas(this, videoControl);
            canvas.addCommand(back);
            canvas.addCommand(capture);
            canvas.setCommandListener(this);
            display.setCurrent(canvas);
            player.start();
        } catch (IOException ioe) {} catch (MediaException me) {}
    }
    .....etc
    class Video extends Thread {
        videoMIDlet midlet;
        public Video(VideoMIDlet midlet) {
            this.midlet = midlet;
        }

        public void run() {
            captureVideo();

        }

        public void captureImage() {
            try {
                byte[] raw = videoControl.getSnapshot(null);
                Image image = Image.createImage(raw, 0, raw.length);
                form.append(image);
                display.setCurrent(form);

                player.close();
                player = null;
                videoControl = null;
            } catch (MediaException me) { }
        }
    };
}
```

```
public class VideoCanvas extends Canvas {
   private VideoMIDlet midlet;

   public VideoCanvas(VideoMIDlet midlet, VideoControl videoControl) {
      int width = getWidth();
      int height = getHeight();
      this.midlet = midlet;

      videoControl.initDisplayMode(VideoControl.USE_DIRECT_VIDEO, this);
      try {
         videoControl.setDisplayLocation(2, 2);
         videoControl.setDisplaySize(width - 4, height - 4);
      } catch (MediaException me) { }
      videoControl.setVisible(true);
   }

   public void paint(Graphics g) {
      int width = getWidth();
      int height = getHeight();

      g.setColor(0x00ff00);
      g.drawRect(0, 0, width - 1, height - 1);
      g.drawRect(1, 1, width - 3, height - 3);
   }
}
```

Pada dua potongan kode hasil implementasi diatas terutama pada kelas *JepretRight* terdapat metod / prosedur *captureImage()*. Prosedur ini akan menghasilkan sebuah *Image* dalam representasi *array of bytes* yang berisi kumpulan warna dari citra hasil *capture*.

Setelah mendapatkan citra dalam bentuk *array of byte* selanjutnya akan dilakukan *watermarking* dengan metode spasial. Metode ini dilakukan dengan mengubah warna dari lokasi *byte* tertentu dimana posisi dari *byte array* yang di *watermarking* diketahui oleh pencipta gambar.

Berikut potongan metode hasil implementasi dari metode spasial *watermarking* dalam J2ME.

```
public Image captureWatermark(byte[] raw, int[] key, byte keyColor) {
   int i=0;

   for(i=0;i < key.Length();i++){
        raw[key[i]] = keyColor;
   }

   Image image = Image.createImage(raw, 0, raw.length);
   return image;
}
```

Image yang dihasilkan dari gambar diatas merupakan citra yang telah mengalami proses *watermarking* secara spasial dengan kunci *key* dan nilai warna yang menggantikan warna pada *pixels* yang mengalami *watermarking* adalah *keyColor*.

Ketika melakukan proses *watermarking* pada kode diatas, hanya terjadi proses pengulangan sebanyak panjang kunci yang diinginkan. Sehingga ukuran dari kecepatan untuk melakukan proses *watermarking* ini tergantung kepada panjang kunci yang digunakan.

Misalkan waktu dalam melakukan *watermarking* dilambangkan dengan simbol *W*, panjang kunci dalam proses ini sejumlah K, dan satuan waktu untuk melakukan satu kali pengulangan adalah n second, maka didapatkan rumus sebagai berikut :

$$W = n.\ K$$

10. Implementasi Ekstraksi *Image Watermarking* dengan J2ME

Proses untuk melakukan Ekstraksi / Deteksi dari sebuah citra yang telah di*watermark* menggunakan metode spasial ini tidak jauh berbeda dengan proses melakukan *watermarking*nya. Hal ini dikarenakan langkah – langkah yang digunakan hanya melakukan pengulangan terhadap kunci yang ada dan mengembalikan ke dalam warna yang semula.

Namun terdapat problematika dalam mengembalikan warna yang semula. Masalah tersebut adalah bagaimana informasi warna yang semula dapat dikembalikan.

Untuk mengatasi permasalahan tersebut maka program harusnya menyimpan informasi itu dalam sebuah file / ataupun basis data RMS yang terdapat pada J2ME.

11. Kualitas Citra Hasil *Watermarking*

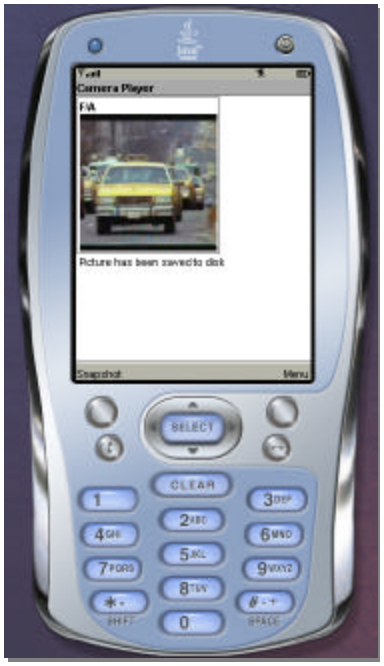
Dalam melakukan proses *digital image watermarking* pada *mobile device*, faktor yang harus tetap diperhatikan adalah kualitas dari citra.

Ketika mengimplementasikan proses *watermarking* memperhatikan panjang kunci dan susunan letak kunci agar kualitas yang dihasilkan tidak jauh berbeda. Pada program sederhana ini belum diimplementasikan informasi mengenai keterhubungan letak *pixels* kunci dengan warna kunci sehingga warna kunci yang ada hanya seragam. Seharusnya warna yang menggantikan yang lebih baik adalah warna yang mendekati warna aslinya.

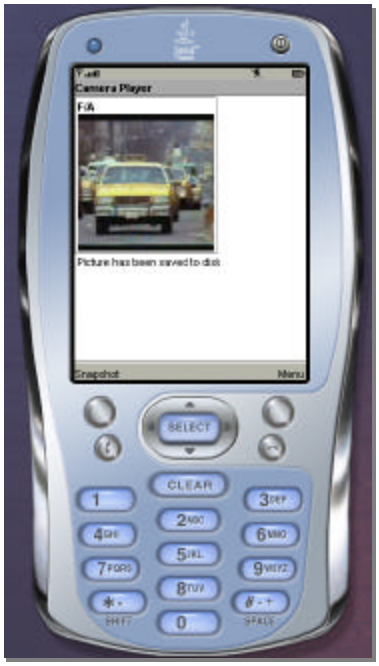
Keterhubungan antara posisi kunci dalam *pixels* dengan warna pengganti dapat diimplementasikan dalam sebuah *vector* dan disimpan secara tetap pada RMS J2ME.

Jika diimplementasikan dalam bahasa C++ dapat digambarkan *vector* yang bisa diimplementasikan adalah sebagai berikut :

```
vector <int posByte,byte color>
```



Gambar 8. Citra Asli 1



Gambar 9. Citra Hasil *Watermark* 1

Gambar 10. Citra Asli 2

Gambar 11. Citra Hasil *Watermark* 2

Gambar 12. Citra Asli 3

Gambar 13. Citra Hasil *Watermark* 3

12. Implementasi Aplikasi *Digital Image Watermarking* pada *nokia 9500*

Untuk menguji penerapan digital image *watermarking* selain dari simulator J2ME, dilakukan pengujian terhadap perangkat *hanphone* nokia 9500.

Pengujian yang dilakukan pada *device* ini dilakukan dengan kode program yang sesederhana mungkin seperti yang dituliskan dalam makalah ini.

Hasil dari pengujian tersebut adalah sebagai berikut :

Gambar 14. Hasil *Capture* Foto dari Nokia 9500

Gambar 10 Hasil *Watermarking*

13. Analisa Hasil dari Nokia 9500

Analisa yang berhasil dilakukan setelah mengimplementasikan program ini pada Nokia 9500 adalah sebagai berikut :

1. Kualitas dari citra hasil *watermarking* mengalami sedikit penurunan. Hal ini dikarenakan warna pengganti yang digunakan seragam.
2. Waktu *W* untuk melakukan proses *watermarking* dapat dikatakan cukup cepat. Hal ini dikarenakan kecepatan dari Nokia 9500 untuk melakukan komputasi.
3. Aplikasi yang di*install* pada Nokia 9500 seringkali tidak berjalan dengan baik. Hal ini dikarenakan terdapat kekurangan dalam melakukan penulisan kode program J2ME (Kemangkusan penggunaan algoritma).

14. Kesimpulan

Kesimpulan yang dapat diambil dari studi dan implementasi *digital image watermarking* pada *mobile device* ini adalah sebagai berikut :

1. *Digital Image Watermarking* dapat diterapkan pada citra / foto yang dihasilkan dari kamera *handphone*.
2. Lama waktu yang diperlukan dalam melakukan proses *watermarking* pada *mobile device* ini tergantung pada panjang kunci K dan n satuan waktu dalam proses pengulangan yang dapat dilakukan oleh *mobile device*. Dirumuskan dengan :

$$W = n.\ K$$

3. Lama waktu yang diperlukan untuk melakukan proses ekstraksi / deteksi dari citra yang telah ter*watermark* sebanding dengan proses *watermarking*.
4. Informasi yang bisa disisipkan dalam metode spasial hanya informasi mengenai letak *pixels* yang akan digantikan warnanya dengan warna kunci.
5. Dalam melakukan proses *watermarking* dengan *mobile device* harus dilakukan penyimpanan terhadap warna asli dari citra yang di*watermark*. Hal ini dikarenakan agar proses dalam melakukan ekstraksi / deteksi kembali dari citra yang ber*watermark* dapat dilakukan dengan mudah.
6. Untuk mempertahankan kualitas dari citra yang telah di*watermark* dapat dilakukan dengan memperpendek panjang kunci dan mengganti warna asli dengan warna yang mendekati. Informasi mengenai keterhubungan letak dan warna pengganti harus disimpan.
7. Kecepatan melakukan proses *watermarking* tergantung dari kemampuan dari *mobile device* untuk melakukan proses komputasi.

Referensi

[1] Kejariwal Arun, Gupta Sumit, Nicolau Alexandru, Dutt Nikil, Gupta Rajesh. *Energy Analysis of Multimedia Watermarking on Mobile Handheld Device.*

[2] P. Mohanty Saraju, *Digital Watermarking*, Departement of Computer Science and Engineering

[3] S.P.Mohanty,et al., *A Dual Watermarking Technique for Images*, *Proc. 7th ACM International Multimedia Conference, ACM-MM'99*, Part 2, pp. 49-51, Orlando, USA, Oct. 1999.

[4] Munir, Rinaldi. (2006). *Diktat Kuliah IF5054 Kriptografi*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[5] V. K. Rohatgi, "An Introduction to Probability Theory and Mathematical Statistics", Wiley Eastern Ltd., 1993.

[6] A.M.Tekalp, "Digital Video Processing", Printice Hall, Englewood Cliffs, NJ, 1995.

[7] David Kahn, "Codebreakers : Story of Secret Writting", Macmillan 1967.

[8] F.L.Bauer, "Decrypted Secrets-Methods and Maxims of Cryptology", Berlin, Heidelberg, Germany: Springer-Verlag, 1997.

[9] Homer, "The Iliad" (trans. R. Fragels), Middlesex, England: Penguin 1972.

[10] Herodotus, "The Histories" (trans. R. Selincourt), Middlesex, England: Penguin 1972.

[11] R. G. Gallager, "Information Theory and Reliable Communication",Wiley, 1968.

[12] J. G. Proakis, "Digital Communications", McGrawhill 1995, 3rd ed.

[13] A. J. Viterbi, "CDMA Principles of Spread Spectrum Communications", Addison-Wesley Inc., 1995.

[14] Rajmohan, "Watermarking of Digital Images", ME Thesis Report, Dept. Electrical Engineering, Indian Institute of Science, Bangalore, India, 1998.

[15] S.P.Mohanty, "Watermarking of Digital Images", Masters Project Report, Dept. of Electrical Engineering, Indian Institute of Science, Bangalore - 560 012, India, Jan 1999.

[16] B.Pfitzmann, "Information Hiding Terminology", *Proc. of First Int. Workshop on Information Hiding*, Cambridge, UK, May30-June1, 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp.347-350.

[17] W. Bendor, et. al., "Techniques for Data Hiding", *IBM Systems Journal*, Vol.35, No.3 and 4, pp. 313-336, 1996.

[18] B.M.Macq and J.J.Quisquater, "Cryptography for Digital TV Broadcasting", *Proc. of the IEEE*, Vol.83, No.6, June 1995, pp. 944-957.

[19] David Kahn, "The History of Steganography", *Proc. of First Int. Workshop on Information Hiding*, Cambridge, UK, May30-June1 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp.1-7.

[20] R.J. Anderson and Fabien A.P. Petitcolas, "On the Limits of Steganography", *IEEE Journal on Selected Areas in Comm*, Vol.16, No.4, May 1998, pp.474-481.

[21] R.J. Anderson, "Stretching the Limits of Steganography", *Proc. of First Int. Workshop on Information Hiding*, Cambridge, UK, May30-June1 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.).

[22] E. Franz, et. al., "Computer Based Steganography", *Proc. First Intl. Workshop on Information Hiding*, Cambridge, UK, May 30 - June 1, 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.).

[23] F.A.P.Petitcolas, et al., "Information Hiding - A Survey", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1062-1078.

[24] C.Cachin, "An Information-Theoritic Model for Steganography", Proc. of the 2nd International Workshop on Information Hiding, Portland, Oregon, USA, 15-17 Apr 1998, Lecture notes in CS, Vol.1525, Springer-Verlag.

[25] S.Craver, "On Public-Key Steganography in the Presence of an Active Warden", Proc. of the 2nd International Workshop on Information Hiding, Portland, Oregon, USA, 15-17 Apr 1998, Lecture notes in Comp Sc, Vol.1525, Springer-Verlag.

[26] N.F.Johnson and Sushil Jajodia, Exploring Steganography: Seeing the Unseen", IEEE Computer, Vol.31,No.2, pp.26-34, feb.1998.

[27] N.Paskin, "Towards Unique Identifiers", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1208-1227.

[28] K.Hill, "A Perspective: The Role of Identifiers in Managing and Protecting Intellectual Property in the Digital Age", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1228-1238.

[29] P.B.Schneck, "Persistent Access Control to Prevent Piracy of Digital Information", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1239-1250.

[30] D.Augot, et al., "Secure Delivery of Images over Open Network", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1251-1266.

[31] M.D.Swanson, et al., "Multimedia data Embedding and Watermarking Technologies", *Proc. of the IEEE*, Vol.86, No.6, June 1998, pp.1064-1087.

[32] Hal Berghel, "Watermarking Cyberspace", *Communications of the ACM*, Nov.1997, Vol.40, No.11, pp.19-24.

[33] M.M.Yeung, "Digital Watermarking", *Communications of the ACM*, Jul.1998, Vol.41, No.7, pp.31-33.

[34] N.Memon and P.W.Wong, "Protecting Digital Media Content", *Communications of the ACM*, July 1998, Vol.41, No.7, pp.35-43.

[35] M.M Yeung, et al. "Digital Watermarking for High-Quality Imaging", *IEEE First Workshop on Multimedia Signal Processing*, June23-25 1997, Princeton, New Jersey, pp. 357-362.

[36] F. Mintzer, et.al., "Effective and Ineffective Digital Watermarks", *IEEE Intl. Conference on Image Processing, ICIP-97*, Vol.3, pp.9-12.

[37] J. Zhao, et. al., "In Business Today and Tommorrow", *Communications of the ACM*, July 1998, Vol.41, No.7, pp.67-72.

[38] J. M. Acken, "How Watermarking Value to Digital Content?", *Communications of the ACM*, July 1998, Vol.41, No.7, pp.75-77.

[39] S. Craver, et. al., "Technical Trials and Legal Tribulations", *Communications of the ACM*, July 1998, Vol.41, No.7, pp.45-54.

[40] I. J. Cox and M. Miller, "A Review of Watermarking and Importance of Perceptual Modelling", *Proc. SPIE Human Vision and Imaging*, SPIE-3016, Feb 1997.

[41] F. Mintzer, et. al., "Opportunities for Watermarking Standards", *Communications of the ACM*, July 1998, Vol.41, No.7, pp.57-64.

[42] G.Voyatzis and I.Pitas, "Protecting Digital Image Copyrights: A Framework", *IEEE Computer Graphics & Applications*, Jan/Feb 1999, pp.18-24.

[43] C.Busch, et al., "D igital Watermarking: From Concepts to Real-Time Video Applications", *IEEE Computer Graphics & Applications*, Jan/Feb 1999, pp. 25-35.

[44] F.Bartolini, et al., "Mask Building for Perceptually Hiding Frequency EmbeddedWatermarks", *Proc. IEEE International Conference on Image Processing, ICIP-98*, Vol.1, pp.450-454.

[45] R.Barnett, "Digital Watermarking : application, techniques, and challengs", *IEE Electronics and Communication Engineering Journal*, August 1999, pp.173-183.

[46] Ton Kalker, et al., "Watermark Estimation Through Detector Analysis", *Proc. IEEE International Conference on Image Processing, ICIP-98*, Vol.1, pp.425 - 429.

[47] F.Mintzer, et al., "Safegaurding Digital Library Contents and Users : Digital Watermarking", D-Lib Magazine, December 1997.

[48] C. F. Osborne, et al., "Image and Watermark Registration for Monochrome and Coloured Images", Digital Image Computing, Technology and Applications, Wellington New Zealand, 1997, pp.59-64

[49] A.Z.Tirkel, et al., "Image and Watermark Registration", Signal Processing, Vol.66, No.3, May 1998, pp.373-384.

[50] Jian Zhao, "Look, Its Not Therae", *BYTE Magazine*, January, 1997, pp.401-407.

[51] F.Hartung and M.Kitter, "Multimedia Watermarking Techniques", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1079-1107.

[52] R.B.Wolfgang, et al., "Perceptual Watermarking for Digital Images and Video", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1108-1126.

[53] G.Voyatzis and I.Pitas, "The Use ofWatermarks in the Protection of Digital Multimedia Products", *Proceedings of the IEEE*, Vol.87, No.7, July 1999, pp.1197-1207.

[54] J.Lacy, et al., "Intellectual Property Systems & Digital Watermarking", *Proc. of the 2nd International Workshop on Information Hiding*, Portland, Oregon, USA, 15-17 Apr 1998, Lecture Notes in Computer Science, Vol.1525, Springer-Verlag, David Aucksmith (Ed.).

[55] F. J. MacWilliam and N. J. A. Sloane, "Pseudorandom Sequences and Arrays", *Proc. of the IEEE*, Vol. 64, No. 12, Dec 1976, pp 1715-1729.

[56] D.V. Sarwate and M. B. Pursley, "Cross-correlation of Pseudorandom and Related Sequences", *Proc. of the IEEE*, Vol.68, No.5, May 1980, pp 593-619.

[57] K. N. Ngan, et. al., "Adaptive Cosine Transform Coding of Images in Perceptual Domain", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol.37, No.11, Nov. 1989, pp.1743-1750.

[58] D. J. Granrath, "The Role of Human Visual Models in Image Processing", *Proc. of the IEEE*, Vol.69, No.5, May 1981, pp.552-561.