STUDI DAN PERBANDINGAN METODOLOGI AUTENTIKASI GAMBAR BERBASIS DIGITAL SIGNATURE

Rezza Mahyudin – NIM : 13505055

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15055@students.if.itb.ac.id

Abstrak

Makalah ini mengulas secara mendalam metodologi berbasis *digital signature* yang digunakan untuk proses autentikasi gambar. Secara umum, authentikasi gambar diperhitungkan sebagai suatu prosedur yang memberi garansi bahwa content gambar tidak berubah, atau paling tidak karakterisitik-karakteristik visual (atau semantik) gambar terjaga setelah memanipulasi gambar secara sederhana seperti kompresi JPEG. Teknik di dalam melakukan autentikasi gambar dibagi menjadi dua macam yaitu teknik *labelling* dan teknik *watermarking*.

Makalah ini bertujuan untuk mengulas secara mendalam kedua metodologi yang digunakan di dalam melakukan autentikasi gambar tersebut untuk kemudian akan dilakukan perbandingan di antara metodologi-metodologi tersebut.

***Kata kunci**: digital signature, watermark, labelling,* autentikasi gambar, *legitimacy,* kriptografi.

1. Pendahuluan

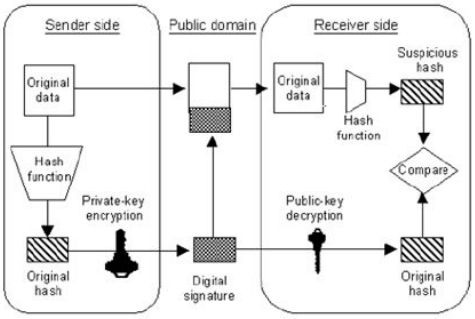
Dewasa ini kemajuan teknologi jaringan komunikasi internasional telah menyediakan fasilitas untuk melakukan pertukaran gambar digital secara efisien. Ironisnya, kemajuan teknologi selama satu dekade terakhir pula yang memungkinkan ketersediaan peralatan sinyal digital atau kakas pemrosesan gambar untuk membuat duplikasi gambar atau bahkan melakukan manipulasi gambar. Hal inilah yang kemudian menyebabkan pentingnya autentikasi gambar dan keutuhan verifikasinya.

Secara umum, autentikasi gambar diperhitungkan sebagai suatu prosedur yang memberi garansi bahwa *content* gambar tidak berubah, atau paling tidak karakterisitik-karakteristik visual (atau semantik) gambar terjaga setelah memanipulasi seperti kompresi JPEG. Dengan kata lain, salah satu tujuan autentikasi gambar adalah untuk membuktikan keutuhan gambar. Untuk beberapa aplikasi seperti pengarsipan medis, pelaporan berita dan kejadian politik, kemampuan deteksi memanipulasi gambar digital seringkali dibutuhkan.

Kebutuhan lain untuk autentikasi gambar datang dari kebutuhan pengecekan identitas pengirim gambar. Pada skenario bahwa seorang pembeli (*buyer*) ingin berbelanja dan menerima sebuah gambar melalui jaringan, pembeli mungkin melihat gambar melalui *e-mail* atau dari *server internet* yang terhubung yang akan memberikan kesempatan jahat pada pihak ketiga (*thirty party*) untuk mencegat dan memanipluasi gambar asli. Untuk itu pembeli membutuhkan kepastian bahwa gambar yang diterima merupakan gambar asli yang dikirim oleh penjual (*seller*). Kebutuhan ini disebut juga sebagai kebutuhan *legitimacy*.

Autentikasi gambar berbasis *digital signature* didasarkan pada konsep *digital signature*, yang diperoleh dari sebuah teknik kriptografi yang disebut *public-key crptosystem*. Gambar 1 Proses *Digital Signature* memperlihatkan model dasar pada *digital signature*. Pengirim pertama-tama menggunakan sebuah fungsi hash seperti MD5 untuk menghasilkan nilai hash data asli (atau *plaintext*) menjadi sebuah file kecil (disebut *diggest*). Lalu *diggest* dienkripsi dengan menggunakan *private key* pengirim. *Diggest* yang telah dienkripsi dapat membentuk sebuah "*signature*" yang unik sebab hanya pengirim

yang mengetahui *private key* yang digunakan. *Signature* lalu dikirim kepada penerima bersama-sama dengan informasi yang asli. Penerima dapat menggunakan *public key* pengirim untuk mendekripsi *signature*, dan menghasilkan *diggest* asli. Tentu saja, informasi yang diterima dapat dihitung nilai hashnya dengan menggunakan fungsi hash yang sama di sisi pengirim. Jika *diggest* yang telah didekripsi tadi cocok dengan *diggest* terbaru yang dibuat, *legitimacy* dan keutuhan pesan sejauh ini terbukti.



Gambar 1. Proses *Digital Signature*

Berdasarkan kebutuhan akan keutuhan gambar (*integrity*) dan kebutuhan *legitimacy*, berbagai variasi teknik telah diusulkan untuk proses autentikasi gambar. Berdasarkan cara yang dipilih untuk menyampaikan data autentikasi, teknik-teknik ini dibagi menjadi dua buah kategori yaitu :

- Teknik berbasis *labelling* (contoh, metode yang diusulkan oleh Friedman, 1993)
- Teknik berbasis *watermarking* (contoh, metode yang diusulkan oleh Walton, 1995).

Perbedaan utama diantara kedua kategori ini adalah bahwa teknik berbasis *labelling* menciptakan data autentikasi atau *signature* yang ditulis ke dalam sebuah file yang terpisah atau sebuah *header* yang dipisahkan dari data mentah yang kemudian tersimpan dalam file yang sama. Sementara teknik berbasis *watermarking* dapat diselesaikan tanpa *overhead* sebuah file yang terpisah atau data autentikasi dilekatkan sebagai *watermark* dalam data mentah itu sendiri.

2. Teknik Berbasis *Labeling*

Pada teknik berbasis *labelling*, informasi autentikasi disampaikan pada sebuah file terpisah yang disebut *label*. Sebuah *label* adalah gabungan informasi tambahan dengan *content* gambar dan dapat digunakan untuk mengidentifikasi gambar. Agar tergabung *content label* dengan *content* gambar, dua cara dapat digunakan dan dinyatakan sebagai berikut :

- Metodologi pertama menggunakan fungsi-fungsi yang biasa digunakan dalam skema autentikasi pesan untuk membangkitkan data autentikasi. Data autentikasi lalu dienkripsi dengan *secret key* atau *private key* bergantung pada protokol autentikasi kriptografi yang digunakan. Saat menggunakannya untuk dua *bit stream* yang berbeda (contoh : data autentikasi yang berbeda), fungsi-fungsi ini dapat memproduksi dua *bit sequence* yang berbeda, seperti suatu cara merubah setiap single bit data autentiksi yang dapat dideteksi. Skema autentikasi gambar di sini ditunjuk sebagai *strict authentication*.
- Metodologi kedua menggunakan beberapa fungsi-fungsi bertujuan khusus (*special-purpose functions*) untuk mengubah karakteristik-karakteristik gambar yang penting dan mengenkripsinya dengan *private key* pengirim. Prosedur ini sama seperti protokol *digital signature* kecuali *feature-feature* yang harus dirancang untuk berkompromi dengan beberapa teknik pengolahan gambar yang spesifik seperti kompresi JPEG. Teknik autentikasi gambar ini dikenal sebagai *non-strict authentication*.

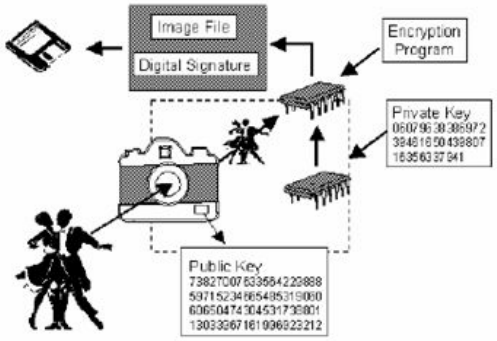
3. Contoh Teknik Berbasis *Labeling*

Penjelasan detil beberapa teknik autentikasi gambar berbasis labeling akan dijelaskan sebagai berikut :

3.1. Teknik Kamera Digital

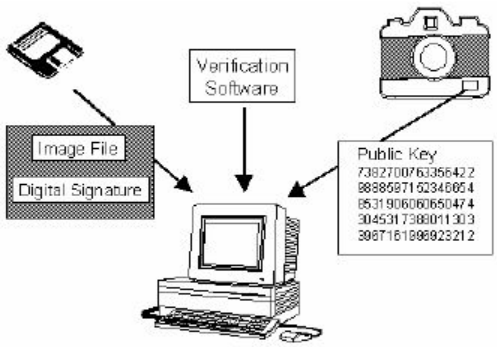
Friedman (1993) menggabungkan ide *digital signature* dengan kamera digital, dan mengusulkan "kamera digital yang dapat dipercaya" diperlihatkan pada gambar 2. Kamera digital yang diusulkan menggunakan sebuah sensor digital sebagai film dan mengirimkam gambar secara langsung dalam format *computer compatible*. *Microprocessor* yang aman diambil untuk ditanamkan di dalam kamera digital dan diprogram dengan *privat key* di pabrik untuk proses enkripsi pada pemberian *digital signature*. *Public key* perlu untuk autentikasi yang hilang nantinya pada *camera body* sebaik batas gambar. Saat kamera digital menangkap gambar objektif, kamera akan memberikan dua *output* file. Salah satunya adalah satu format file standard semua industri digital yang mewakili gambar yang tertangkap, yang lainnya adalah file *digital*

signature yang dienkripsi dengan menggunakan *private key* unik kamera untuk menghitung sebuah nilai hash pada file gambar yang tertangkap. File gambar digital dan *digital signature* nantinya dapat didistribusikan secara bebas dan aman.



Gambar 2. Teknik Kamera Digital

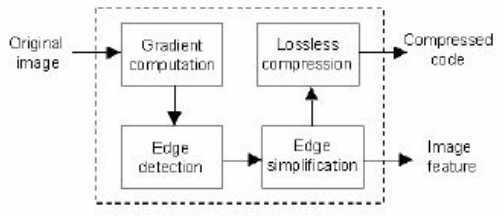
Proses verifikasi ide Friedman diilustasikan pada gambar 3. Autentikasi gambar disempurnakan dengan bantuan software verifikasi *public domain*. Untuk membuktikan sebuah file gambar digital, gambar digital yang menyertai file *digital signature* dan *public key* dibutuhkan ketika menjalankan aplikasi untuk melakukan verifikasi pada platform komputer standard. Program lalu menghitung hash gambar input dan menggunakan *public key* untuk membaca sandi *digital signature* untuk menyatakan hash asli. Jika kedua buah nilai hash cocok, gambar dipertimbangkan untuk memiliki keaslian. Jika nilai dua hash berbeda, keutuhan gambar ini dipertanyakan.



Gambar 3. Verifikasi Teknik Kamera Digital

3.2. Edge-Based Methods

Pinggir / *edge* sebuah gambar adalah batas atau garis bentuk yang memperlihatkan perubahan aspek fisik pada sebuah gambar. *Edges* termasuk salah satu tampilan utama pada sebuah gambar. Dengan menggunakan sebuah peta *binary*, dapat disisipkan *digital signature* pada pinggir / *edge* dari sebuah gambar sebagai skema autentikasi gambar tersebut. Penggunaan teknik ini ikut memperhatikan kemungkinan dilakukannya modifikasi terhadap *edge* (baik posisi, nilai, dan hasil gambar *binary*) ketika rasio kompresi tinggi diterapkan pada gambar. Oleh karena itu, keberhasilan skema teknik *edge-based method* sebagai skema autentikasi sangat bergantung pada kapasitas sistem untuk mengenali dan membedakan *edges* yang dihasilkan oleh manipulasi-manipulasi yang menghadirkan *digital signature*. Proses menyisipkan digital signature pada *edges* sebuah gambar ditunjukkan pada gambar 4 proses *edge-based methods*.



Gambar 4. Proses *Edge-Based Methods*

4. Implementasi *Labelling Digital Signature*

Berikut akan ditampilkan implementasi sederhana fungsi yang menghasilkan *label digital signature* yang kemudian dimasukkan ke dalam file *input* yang sama dengan menggunakan bahasa C#.

```
private void
bukaDokumenToolStripMenuItem_Click(object
sender, EventArgs e)
{
  openFileDialog1.Title = "Buka Dokumen";
  if
(openFileDialog1.ShowDialog().Equals(Dial
ogResult.OK))
  {
     //open file
     CurrentFile =
openFileDialog1.OpenFile();
     FileReader = new
StreamReader(CurrentFile);
     //assign the readresult to current
file bytes
     CurrentFileBytes = new
Byte[CurrentFile.Length];
     CurrentFile.Position = 0;
     int i = 0;
     while (CurrentFile.Position <
CurrentFile.Length)
     {
        if (CurrentFile.Length -
CurrentFile.Position < 8)
        {
```

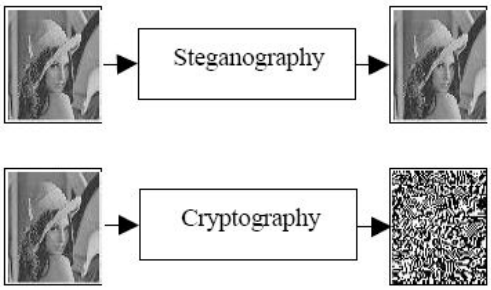
```
            i +=
CurrentFile.Read(CurrentFileBytes, i,
(int)(CurrentFile.Length -
CurrentFile.Position));
         }else{
            i +=
CurrentFile.Read(CurrentFileBytes, i, 8);
         }
      }
}

private void
pemberianTandaTanganToolStripMenuItem_Cli
ck(object sender, EventArgs e)
{
   if (kunciExist == true)
   {
      //create MD5 class
      MyMD5 hash = new
MyMD5(CurrentFileBytes);
      hashresult = hash.ComputeHash();
      //create RSA class
      RSA rsa = new RSA(kunci,hashresult,
16);
      BigInteger encryptresult =
rsa.encrypt();
   }else{
      MessageBox.Show("Kunci Belum
Dibangkitkan");
   }
}

public class MyMD5
{
   private byte[] md5byte;

   /// Konstruktor dengan parameter
berupa namafile
   public MyMD5(byte[] input)
   {
      md5byte = input;
   }

   /// Fungsi utama yang akan dipanggil
untuk melakukan komputasi nilai hash
   public string ComputeHash()
   {
      MD5 md5 = new
MD5CryptoServiceProvider();

      /*Menghitung nilai hash dari isi
file dokumen*/
      Byte[] result =
md5.ComputeHash(md5byte);

      /*Convert nilai hash ke string*/
      String result2 = "";
      for (int j = 0; j < result.Length;
j++)
      {
         result2 += result[j].ToString();
      }

      return result2;
   }
}

public class RSA
{
   private KeyGen kunci;
   private BigInteger document;

   /// kontruktor kelas RSA
   public RSA(KeyGen key, string doc, int
i)
   {
      kunci = key;
      document = new BigInteger(doc,i);
   }

   /// Fungi yang menghasilkan nilai SK
pada algoritma RSA
   public Hashtable GetPrivate()
   {
      Hashtable tmp = new Hashtable();
      tmp.Add(kunci.GetN(),
kunci.GetPrivateKey());
      return tmp;
   }

   /// Fungi yang menghasilkan nilai PK
pada algoritma RSA
   public Hashtable GetPublic()
   {
      Hashtable tmp = new Hashtable();
      tmp.Add(kunci.GetN(),
kunci.GetPublicKey());
      return tmp;
   }

   /// Fungi yang mengimplementasi
persamaan enkripsi pada algoritma RSA
   public BigInteger encrypt()
   {
      IDictionaryEnumerator en =
GetPrivate().GetEnumerator();
      en.MoveNext();
      return
document.modPow((BigInteger)en.Value,
(BigInteger)en.Key);
   }

   /// Fungi yang mengimplementasi
persamaan dekripsi pada algoritma RSA
   public BigInteger decrypt()
   {
      IDictionaryEnumerator en =
GetPublic().GetEnumerator();
      en.MoveNext();
      return
document.modPow((BigInteger)en.Value,
(BigInteger)en.Key);
   }
}
```

5. Teknik Berbasis *Watermarking*

Watermarking merupakan salah satu bentuk dari steganografi (Ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain). *Watermarking* juga dapat dikatakan sebagai ilmu yang mempelajari teknik-teknik penyimpanan suatu data (digital) kedalam data *host* digital yang lain (Istilah *host* digunakan untuk data/sinyal digital yang ditumpangi.).

Steganografi memiliki hasil keluaran (*output*) yang berbeda dengan kriptografi. Hasil dari

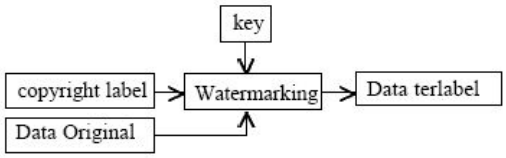
proses kriptografi biasanya berupa data yang berbeda dari bentuk aslinya sementara hasil keluaran dari steganografi memiliki bentuk persepsi yang sama dengan bentuk aslinya apabila ditangkap oleh indera manusia biasa.



Gambar 5. Perbedaan Steganografi dan Kriptografi

Sehingga dapat dikatakan watermarking merupakan suatu cara untuk penyembunyian atau penanaman data / informasi tertentu (baik hanya berupa catatan umum maupun rahasia) ke dalam suatu data digital lainnya, tetapi tidak diketahui kehadirannya oleh indera manusia (indera penglihatan atau indera pendengaran), dan mampu menghadapi proses-proses pengolahan sinyal digital sampai pada tahap tertentu.

6. Proses Watermarking



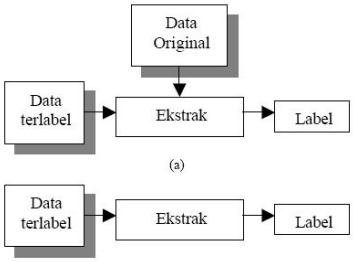
Gambar 6. Proses *Watermarking*

Pada gambar proses *watermarking* diatas, terdapat komponen [*key*], key ini digunakan untuk mencegah penghapusan secara langsung oleh pihak tak bertanggung-jawab, dengan menggukan metoda enkripsi yang sudah ada. Sedangkan ketahanan terhadap proses-proses pengolahan lainnya, itu tergantung pada metoda *watermarking* yang digunakan.

Terdapat 3 sub-bagian yang membentuk sebuah sistem *watermarking*. Ketiga sub-bagian tersebut antara lain :

1. Penghasil Label Watermark
2. Proses penyembunyian Label
3. Menghasilkan kembali Label Watermark dari data yang terwatermark

Label *watermark* adalah sesuatu data/informasi yang akan kita masukkan ke dalam data digital yang ingin di-*watermark.* Secara umum, terdapat dua cara di dalam melakukan ekstraksi label dari sebuah data terlabel. Kedua cara tersebut dapat dilihat pada gambar 7 berikut ini,



Gambar 7. Proses Ekstrak Dengan Data Asli dan Tanpa Data Asli

7. Contoh Teknik Berbasis *Watermarking*

Terdapat banyak metoda *watermarking* untuk citra digital yang sudah diteliti. Ada yang bekerja pada domain spasial atau waktu, dan ada yang mengalami transformasi terlebih dahulu (seperti DCT, FFT, dsb) misalnya ke domain frekuensi. Bahkan ada yang menerapkan teknologi-teknologi lain seperti *fraktal, spread spectrum* untuk telekomunikasi dan sebagianya. Beberapa metoda yang pernah diteliti, diantaranya adalah :

7.1. LSB (*Least Significant Bit*) *Coding*

Metoda ini merupakan metoda yang paling sederhana tetapi yang paling tidak tahan terhadap segala proses yang dapat mengubah nilai-nilai intensitas pada citra. Metoda ini akan mengubah nilai LSB (*Least Significant Bit*) komponen luminansi atau warna menjadi bit yang bersesuai dengan bit label yang akan disembunyikan. Memang metoda ini akan menghasilkan citra rekontruksi yang sangat mirip dengan aslinya, karena hanya mengubah nilai bit terakhir dari data. Tetapi sayang tidak tahan terhadap proses-proses yang dapat mengubah data citra terutama kompresi JPEG. Metoda ini paling mudah diserang, karena bila orang lain tahu maka tinggal membalikkan nilai dari LSB-nya maka data label akan hilang seluruhnya.

7.2. Patchwork

Metoda ini diusulkan oleh Bender. Metoda ini menanamkan *label* 1 bit pada citra digital dengan menggunakan pendekatan statistik. Dalam metoda ini, sebanyak n pasang titik (a_i,b_i) pada citra dipilih secara acak. *Brightness* dari a_i dinaikkan 1 (satu) dan *brightness* dari

pasangannya b_i diturunkan satu. Nilai Harapan dari jumlah perbedaan n pasang titik tersebut adalah 2n. Ketahanan metoda ini terhadap kompresi JPEG dengan parameter kualitas 75%, maka *label* tetap dapat dibaca dengan probabilitas kebenaran sebesar 85%.

7.3. Pitas and Kaskalis

Pitas dan Kaskalis mengusulkan metoda yang hampir sama dengan metoda yang diusulkan oleh Bender. Metoda ini membagi sebuah citra atas dua bagian (*subsets*) sama besar (misalnya dengan menggunakan *random generator*) atau dengan sebuah *digital signature* S yang merupakan pola biner dengan ukuran N x M dimana jumlah *biner* "1" (satu) sama dengan jumlah *biner* "0" (nol). Kemudian salah satu *subset* ditambahkan dengan faktor k (bulat positif). Faktor k diperoleh dari perhitungan variansi dari kedua *subset*. Verifikasi dilakukan dengan menghitung perbedaan rata-rata antara kedua *subset*. Nilai yang diharapkan adalah k bila ada *label* yang ditanamkan. Metoda ini hanya tahan terhadap kompresi JPEG dengan ratio 4:1 (faktor kualitas kira-kira lebih dari 90%).

7.4. Caroni

Caroni mengusulkan metoda penyembunyian sejumlah bit *label* pada komponen luminansi dari citra dengan membagi atas blok-blok, kemudian setiap pixel dari satu blok akan dinaikan dengan faktor tertentu bila ingin menanamkan bit '1', dan nilai-nilai pixel dari blok akan dibiarkan bila akan menanamkan bit '0'. Untuk mendapatkan *label*nya kembali, maka *brightness* setiap titik dari citra yang terlabel akan dikurangkan dengan citra asli. Jika rata-rata dari satu blok pixel melewati suatu nilai (*threshold*) tertentu, maka akan dinyatakan sebagai bit '1', bila tidak maka dinyatakan sebagai bit '0'. Setelah mengalami kompresi JPEG, metoda ini dapat tahan terhadap faktor kualitas sebesar 30%.

8. Implementasi *Watermarking*

Berikut akan ditampilkan implementasi sederhana fungsi yang menghasilkan *watermarking* dalam bahasa C#.

```
/* Display the watermark as it would
appear after the watermark were saved to
the file */
private void btnPreview_Click(object
sender, EventArgs e)
{
   // Update the application by reloading
the image
   picContainer.Image =
Image.FromFile(CurrentFile);

   int opac = 0;
   string sOpacity = cboOpacity.Text;

   // Determine the opacity of the
watermark
   switch (sOpacity)
   {
      case "100%":
         opac = 255; // 1 * 255
         break;
      case "75%":
         opac = 191; // .75 * 255
         break;
      case "50%":
         opac = 127; // .5 * 255
         break;
      case "25%":
         opac = 64;  // .25 * 255
         break;
        case "10%":
           opac = 25;  // .10 * 255
           break;
        default: // default at 50%
           opac = 127;// .5 * 255
           break;
     }

   // Get a graphics context
   Graphics g =
Graphics.FromImage(picContainer.Image);

   // Create a solid brush to write the
watermark text on the image
   Brush myBrush = new
SolidBrush(Color.FromArgb(opac,
            myWatermarkColor));

   // Calculate the size of the text
   SizeF sz =
g.MeasureString(txtWaterMark.Text,
myFont);

   // Create a copy of variables to keep
track of the drawing position (X,Y)
            int X;
            int Y;

   // Set the drawing position based on
the users selection of placing the text
at the bottom or top of the image
   if (optTop.Checked == true)
    {
       X = (int)(picContainer.Image.Width
- sz.Width) / 2;
       Y = (int)(picContainer.Top +
sz.Height) / 2;
    }else{
       X = (int)(picContainer.Image.Width
- sz.Width) / 2;
       Y =
(int)(picContainer.Image.Height -
sz.Height);
    }

   // Draw the water mark text
   g.DrawString(txtWaterMark.Text,
myFont, myBrush, new Point(X, Y));
```

```
    // Set the font and color of the font
for the watermark
   private void btnFont_Click(object
sender, EventArgs e)
   {
     // default the current font and
color to that used in the watermark
textbox
     fontDialog1.ShowColor = true;
     fontDialog1.Font=txtWaterMark.Font;
     fontDialog1.Color =
txtWaterMark.ForeColor;

     if (fontDialog1.ShowDialog() !=
DialogResult.Cancel)
     {
        myFont = fontDialog1.Font;
        myWatermarkColor =
fontDialog1.Color;
        txtWaterMark.Font =
fontDialog1.Font;
        txtWaterMark.ForeColor =
fontDialog1.Color;
     }
}
```

9. Perbandingan *Labelling* dan *Watermarking*

9.1. Kelemahan

Kelemahan yang dimiliki oleh teknik *labelling* dalam autentikasi gambar berbasis *digital signature* adalah sebagai berikut,

1. Pada teknik berbasis *labelling*, penyimpanan *digital signature* dalam sebuah file terpisah, atau dalam segment *header* yang terpisah pada file yang berisi data mentah memunculkan pentingnya *maintenance overhead* dan mungkin memberikan transmisi ekstra pada file-file *signature*.
2. Saat *signed medium* dimanipulasi, *signature* yang disimpan tidak subjektif untuk proses manipulasi yang sama, memberikan kesulitan untuk menduga apakah manipulasi telah dilakukan dan untuk menunjuk lokasi sementara (*temporal*) dan ruang yang renggang (*spatial*) dimana kerusakan terjadi.
3. Penggunaan *digital signatures* tradisional untuk *labelling* tidak cocok untuk aplikasi *lossy* atau *progressive transmission*.
4. *Transcoding* atau *converting format* pada media yang dilindungi dengan teknik berbasis *labelling* tidak selalu tepat.

Sementara itu, kelemahan yang dimiliki oleh teknik *watermarking* dalam autentikasi gambar berbasis *digital signature* adalah data yang digunakan sebagai *digital signature* tidak dapat menggunakan data teks. Hal ini disebabkan perubahan 1 bit pada data tersebut akan menghasilkan teks yang jauh berbeda. Selain itu, penyisipan *digital signature* sebagai *watermark* pada gambar dapat mengurangi / merubah tampilan asli dari gambar.

9.2. Kelebihan

Kelebihan yang dimiliki oleh teknik *labelling* hampir sama dengan kelebihan yang dimiliki oleh teknik *watermarking*. Kelebihan-kelebihan yang dimiliki oleh kedua teknik tersebut adalah,

1. Keduanya dapat mendeteksi perubahan setiap single bit data image jika keutuhan yang sempurna (*strict integrity*) harus dijamin.
2. Authentikasi gambar dapat dilakukan dalam cara keamanan dan *robust* di *public domain* (contoh : internet)
3. Kapasitas data tersembunyi pada teknik berbasis *labelling* lebih tinggi dibandingkan *watermaking*.

10. Kesimpulan

Berdasarkan aplikasi yang dibahas pada laporan ini maka teknik autentikasi dibagi atas teknik berbasis *labelling* dan teknik berbasis *watermaking*. Setiap teknik memiliki kelemahan dan kekurangan masing-masing. Baik teknik *labelling* maupun teknik *watermarking* masih dapat diteliti dan dikembangkan untuk mendapatkan hasil *digital signature* yang lebih baik lagi. Namun, untuk saat ini, teknik *watermarking* masih dianggap sedikit lebih baik daripada teknik *labelling* sebagai metodologi autentikasi gambar berbasis *digital signature*.

11. Daftar Pustaka

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Lu, Chu-Shien. (2005). Multimedia Security : Steganography and Digital Watermaking Techniques for Protection of Intelectual Property. Idea Group Publishing.

[3] Lin, Ching-Yung; Chang, Shih-Fu. (2009). Generating Robust Digital Signature for Image/Video. http://www.ee.columbia.edu