DIGITAL RIGHTS MANAGEMENT DAN VARIASINYA : MENUJU ERA *GAME* TANPA PEMBAJAKAN

Shieny Aprilia – NIM : 13505089

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jln. Ganesha 10, Bandung
E-mail : if15089@students.if.itb.ac.id

Abstrak

Makalah ini akan membahas *Digital Rights Management*, yaitu teknologi yang digunakan untuk melindungi karya intelektual yang berbentuk digital, seperti musik, video, film, perangkat lunak, *game*, dan sebagainya. Sistem DRM harus mampu mencapai beberapa tujuan untuk dapat menjadi semacam *platform* yang dapat dipercaya oleh masyarakat yang ingin terjun ke dunia *e-market*. Selain itu, makalah ini akan membahas arsitektur dan teknik-teknik DRM yang umum digunakan.
Untuk game, telah ada beberapa *framework* khusus yang umum digunakan untuk melindungi HaKI (Hak atas Kekayaan Intelektual) atas game. Namun, sampai saat ini pembajakan game masih sering terjadi, khususnya di Indonesia. Oleh karena itu, dibutuhkan suatu teknik baru yang kuat terhadap serangan-serangan yang mungkin ditujukan untuk membobol DRM terhadap game.

Kata Kunci: *Digital Rights Management*, game, enkripsi, dekripsi, kunci, *one time pad*, *vernam algorithm*.

1. Pendahuluan

Pada jaman dahulu, media hiburan seperti musik, film, dan sebagainya, didistribusikan dengan menggunakan benda analog semacam piringan hitam dan kaset video. Namun, seiring perkembangan jaman, hampir semua media hiburan sudah didistribusikan dengan menggunakan benda digital semacam CD (*Compact Disc*) dan *file* yang bisa bebas diunduh dari internet, karena kualitas benda digital jauh lebih bagus daripada benda analog. Benda analog akan mengalami depresiasi kualitas seiring dengan semakin seringnya benda tersebut di-*copy*, sedangkan hasil salinan yang sama dapat dengan mudah diperoleh jika benda yang disalin tersebut adalah benda digital. Namun, penggunaan benda digital ini membawa dampak positif dan negatif. Dampak positifnya adalah bahwa benda digital dapat dengan mudah di-*copy*, dikirimkan, dan disebarkan melalui jaringan komputer. Perkembangan internet yang sangat pesat membuatnya menjadi sarana paling baik untuk bertukar informasi sekarang ini. Selain itu, kakas pertukaran *file* membuat penyebaran karya digital yang dilindungi menjadi semakin mudah dan cepat. Sedangkan dampak negatifnya adalah HaKI (Hak atas Kekayaan Intelektual) atas benda digital tersebut semakin sulit untuk dilindungi, mengingat benda digital hasil salinan dan yang orisinil tidak bisa dibedakan satu sama lain.
Untuk itulah, diperlukan suatu mekanisme yang bisa menjaga lisensi informasi digital yang dapat dengan mudah tersebar luas melalui jaringan internet. Mekanisme ini disebut *Digital Rights Management* (DRM).

2. *Digital Rights Management*

Secara umum, *Digital Rights Management* (DRM) adalah kumpulan teknologi yang secara teknis dapat digunakan untuk menjamin lisensi informasi digital. Dengan DRM, pendistribusian informasi berharga secara digital menjadi mungkin untuk dilakukan tanpa melanggar HaKI. Sistem DRM dirancang untuk melindungi informasi secara persisten sesuai dengan aturan komersial media yang dilindungi tersebut.
DRM sangatlah penting bagi para pembuat dan pemasar media elektronik karena DRM membantu mereka untuk menjaga keuntungan akan produk yang mereka pasarkan. Sejak perkembangan komputer personal yang semakin pesat, peng-*copy*-an media digital menjadi semakin mudah. Media digital dapat disalin sesering mungkin tanpa mengurangi kualitasnya. Dengan mengendalikan pertukaran media digital tersebut, DRM membantu para pemasar media digital untuk membatasi sirkulasi media digital secara ilegal. Umumnya, DRM melindungi suatu media digital dengan melakukan enkripsi data sehingga media tersebut hanya dapat diakses oleh pihak-pihak yang berkepentingan atau dengan melakukan *digital watermarking* sehingga media tersebut tidak dapat disebarluaskan secara bebas. Metode manapun yang digunakan, DRM memastikan suatu media digital hanya digunakan oleh pihak-pihak yang

telah membayar untuk itu dan/atau mempunyai hak untuk mengakses media tersebut.
DRM terkadang diartikan sebagai *Digital Restriction Management* oleh pihak-pihak yang memiliki pandangan atau pengalaman buruk mengenai DRM. Pihak-pihak tersebut merasa bahwa DRM tidak lebih dari sistem yang melarang mereka melakukan hal-hal yang merupakan hak mereka atau melanggar privasi mereka dengan permintaan untuk memasukan identitas mereka sebelum melakukan pengaksesan data.

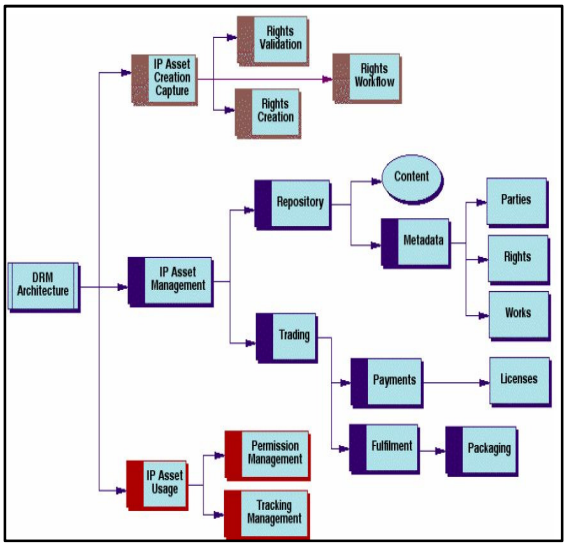
2.1 Tujuan DRM

DRM harus mampu mencapai beberapa tujuan tertentu agar dapat menciptakan kondisi yang dapat dipercaya oleh pihak-pihak yang ingin terlibat dalam *e-market*. DRM akan memungkinkan terjadinya penyebaran media digital komersial yang legal dengan mencapai tujuan-tujuan berikut ini:

1. DRM harus dapat menyediakan perlindungan terhadap media digital
2. DRM harus dapat memungkinkan penyebaran media digital yang aman
3. DRM harus dapat memastikan keabsahan suatu media digital
4. DRM harus dapat menjamin terjadinya transaksi yang legal
5. DRM harus dapat menyediakan identitas pihak yang terlibat dalam *e-market*

2.2 Arsitektur DRM

Pada implementasinya, tidak ada arsitektur DRM yang standard, karena biasanya setiap perusahaan menawarkan *framework* DRM yang berbeda dari perusahaan yang lain. Namun secara umum arsitektur *framework* DRM adalah seperti yang dapat dilihat pada Gambar 1.



Gambar 1 Arsitektur DRM

Arsitektur *framework* DRM dapat dibedakan menjadi 3 bagian :

1. *Intellectual Property (IP) Asset Creation and Capture*
   Mencakup proses pengelolaan dan pembuatan karya untuk mempermudah proses distribusinya, termasuk juga membuat lisensi karya tersebut ketika pertama kali dibuat.
2. *IP Asset Management*
   Proses yang mengelola pertukaran karya digital, termasuk menerima karya sebagai input untuk sistem manajemen karya digital.
3. *IP Asset Usage*
   Mencakup modul yang mengendalikan penggunaan karya digital setelah sampai ke tangan pengguna, termasuk batasan-batasan pertukaran media digital dari satu komputer ke komputer lainnya.

2.3 Teknik DRM

Teknik-teknik yang umum digunakan pada sistem DRM adalah sebagai berikut:

1. Enkripsi
   DRM menggunakan algoritma kriptografi untuk melakukan enkripsi data media digital dengan menggunakan kunci rahasia yang hanya diumumkan kepada pihak-pihak yang berhak saja. Dengan demikian hanya pihak-pihak yang berhak saja yang dapat melakukan dekripsi dan kemudian membaca media digital tersebut.
   Hal yang terpenting dalam penggunaan algoritma kriptografi ini adalah cara pembuatan kunci dekripsinya, cara pengiriman kunci tersebut ke pihak yang berhak, cara untuk membatasi waktu penggunaan media digital, dan cara untuk mencegah pencurian kunci dekripsi.
2. Kunci publik/privat
   Teknik ini masih termasuk ke dalam teknik kriptografi yang menggunakan dua kunci yang berbeda, yang disebut kunci publik dan privat. Setiap kunci tersebut dapat digunakan untuk melakukan enkripsi atau dekripsi data. Teknik kriptografi ini disebut sebagai kriptografi asimetrik. Kriptografi asimetrik sangatlah kuat karena dapat menyediakan fungsionalitas yang menambah kepercayaan akan sistem keamanan media, seperti *digital signatures*. Namun, teknik ini membutuhkan komputasi yang kompleks. Oleh karena itu, pada sistem seperti SSL, sistem DRM mengkombinasikannya dengan kriptografi simetrik yang dikenal dengan nama kriptografi kunci rahasia, yaitu penggunaan kunci yang sama untuk proses enkripsi maupun dekripsi.
3. *Digital Certificates*

Seperti pada kehidupan nyata di mana seseorang harus menyatakan identitasnya pada suatu transaksi pembayaran, seseorang pun harus membuktikan kebenaran identitas virtualnya di *e-market* dengan menggunakan *digital certificate*. Dengan kata lain, *digital certificate* adalah sarana penghubung antara identitas seseorang yang sebenarnya dengan identitas virtualnya, yang dibuat dengan menggunakan teknik kriptografi yang menghubungkan identitas orang tersebut dengan kunci kriptografi publiknya. Layanan ini disediakan oleh *certificate authorities* yang menjamin bahwa suatu kunci publik adalah milik seseorang yang namanya tercantum pada sertifikatnya.

4. *Watermarking*
   Proses ini menyembunyikan informasi pada suatu sumber data. Penggunaannya biasanya untuk menyimpan cap lisensi sebuah media pada media itu sendiri. Proses *watermarking* harus menjamin bahwa media tempat persembunyian informasi tersebut tidak mengalami penurunan kualitas yang dapat dideteksi oleh indra visual atau audio manusia. Dengan demikian hanya informasi mengenai kunci rahasia *watermarking* yang dapat mengekstrak informasi rahasia yang tersembunyi tersebut.
5. Kontrol terhadap akses
   Sistem DRM bukan hanya harus mampu menyediakan mekanisme pencegahan pengkopian media, melainkan harus mampu memastikan bahwa media digital tersebut hanya dapat diakses oleh pihak yang berhak saja. Caranya adalah dengan menggunakan teknik enkripsi data.
6. Protokol komunikasi yang aman
   Secure Sockets Layer (SSL) dan *Transport Layer Security* (TLS) adalah protokol kriptografi yang menyediakan mekanisme komunikasi yang aman melalui internet. Protokol ini memungkinkan aplikasi *client*/*server* untuk berkomunikasi dengan aman tanpa adanya pihak yang ikut memperoleh informasi (*eavesdropping*). IP *Security* (IPsec) adalah standar pada *network layer* yang berfungsi untuk mengamankan komunikasi IP dengan cara melakukan enkripsi dan/atau autentikasi terhadap semua paket IP.
7. *Fingerprinting*
   Fingerprinting atau penggunaan sidik jari melindungi media digital dengan menyelipkan sidik jari si pembeli pada media tersebut. Metode ini mirip dengan *watermarking*.
8. *Rights specification language*
   Metode ini menyediakan mekanisme untuk mendeskripsikan hak pembuat atau pemasar media digital.
9. Infrastruktur yang dapat dipercaya
   Infrastruktur yang terlibat dalam proses pengiriman, *opening*, *displaying*, dan *disposing* pemaketan media digital dari pemasar sampai ke konsumen harus menyediakan layanan yang dapat dipercaya oleh para konsumen.
10. *Hashing*
    DRM dapat melindungi media digital dari praktek manipulasi dengan menggunakan *one-way hash function*. Masukan fungsi ini berupa media digital dan keluarannya berupa *message digest*. Perubahan sedikit saja pada sebuah media digital akan menghasilkan *message digest* yang sangat berbeda pula. Oleh karena itu, jika seorang konsumen ingin mengecek keabsahan suatu media digital, maka yang perlu dilakukannya hanyalah menggunakan *one-way hash function* ini untuk melihat apakah *message digest* yang dihasilkannya sama dengan yang diberikan oleh pembuat aslinya.

3. DRM dan *Video Game*

Saat ini, salah satu media hiburan bernilai komersial tertinggi di dunia adalah *video game* (untuk selanjutnya akan disebut sebagai "game" saja). Namun, seperti media hiburan lainnya, game juga tidak terlepas dari praktik pembajakan. Khususnya di Indonesia, pembajakan game bukan merupakan hal yang aneh lagi. Hampir semua penikmat game di Indonesia menikmati game secara ilegal, yaitu mendapatkannya melalui praktek pembajakan. Praktek pembajakan ini bisa melalui pembelian game bajakan, penggunaan program *keygen* untuk membangkitkan kode serial yang diperlukan untuk menghilangkan sistem DRM suatu game, atau penggunaan game yang telah di-*patch* dengan kode tambahan yang juga bisa menghilangkan sistem DRM game tersebut (biasa disebut sebagai game yang telah di-*crack*). Sistem DRM pada game biasanya berupa sistem yang dimasukkan ke dalam CD game atau semacam kode yang ditambahkan pada game itu sendiri. Sistem DRM pada game bisa digunakan untuk membatasi waktu percobaan game (pada *trial version*) atau untuk membatasi siapa saja yang boleh meng-*install* game tersebut atau membatasi fitur-fitur tertentu tidak dapat digunakan pada *trial version*.

Walaupun teknik kriptografi yang digunakan oleh sistem DRM game sudah diperbarui terus-menerus, sistem tersebut masih bisa dibobol oleh orang-orang yang tidak bertanggung jawab namun cerdik. Oleh karena itu, perlu dirancang suatu teknik DRM

yang meliputi suatu algoritma kriptografi yang kuat untuk membangun sistem DRM yang dapat menahan serangan-serangan terhadap perlindungan lisensi game. Namun, perlu diperhatikan pula bahwa teknik dan algoritma kriptografi yang digunakan harus bebas dari kesalahan, karena kesalahan pada sistem DRM bisa membuat DRM itu sendiri menjadi senjata makan tuan. Hal ini terjadi pada tahun lalu, ketika adanya kesalahan pada sistem DRM SecuROM yang digunakan pada sebuah game populer produksi EA Games, Spore, yang mengakibatkan game tersebut menjadi game yang paling banyak dibajak. Dari kesalahan ini dapat disimpulkan bahwa teknik DRM yang digunakan untuk melindungi suatu game tidak boleh merugikan pihak-pihak yang telah secara legal memperoleh suatu game.

3.1 DRM pada *Casual Game*

Salah satu jenis game yang tingkat pembajakannya merupakan yang paling tinggi dibandingkan dengan jenis game yang lain adalah *casual game*. *Casual game* adalah game yang dapat dikategorikan game ringan, ditujukan bagi semua orang, bukan hanya kalangan yang kegemarannya bermain game, dan penyebarannya melalui internet. Karena perkembangannya yang sangat terikat kuat dengan adanya internet, pembajakan bukan merupakan sesuatu yang asing lagi dalam industri *casual game*. Apalagi karena biasanya para pemasar *casual game* hanya memiliki sebuah sistem DRM tunggal bagi semua *casual game* produksinya, cukup sebuah game saja yang dibobol oleh si pembajak, maka si pembajak akan dapat memainkan semua game milik pemasar yang sama dengan gratis. Para pemasar menggunakan sistem DRM tunggal ini dengan tujuan agar ia lebih mudah dalam mengelola sistem keamanan semua *casual game* produksinya, mengingat sebuah pemasar mungkin saja merilis sebuah *casual game* setiap harinya, sehingga akan banyak sekali game yang harus dikelolanya. Dan jika sistem keamanan untuk setiap game berbeda satu sama lainnya, dapat dibayangkan betapa rumitnya sistem manajemen pemasar tersebut. Oleh karena itu, perlu dipikirkan suatu teknik baru bagi sistem DRM game yang ampuh untuk semua game namun tahan akan serangan para pembajak.

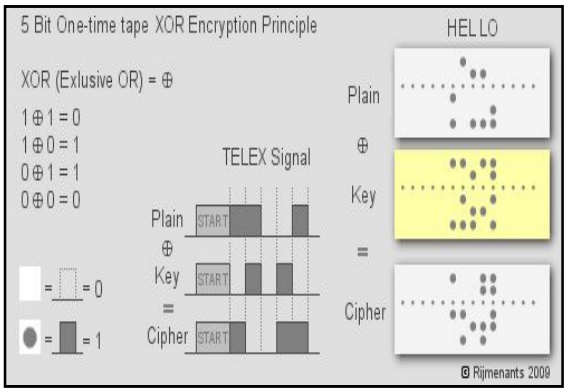
3.2 Cara Penyerangan DRM

Secara umum, terdapat 3 cara yang dapat dipilih para pembajak untuk menyerang DRM suatu game:

1. Eksploitasi
   Cara ini merupakan cara yang paling mudah dibandingkan dengan cara yang lain. Praktek yang termasuk ke dalam cara ini adalah, menghapus batas waktu *trial version* pada *registry* komputer, mengganti nama file .exe yang tersembunyi, atau menggunakan *task manager* untuk menghentikan proses sistem DRM yang sedang berjalan.
2. *Keygen*
   Kebanyakan sistem DRM game, khususnya pada *casual game*, menggunakan sistem enkripsi yang membatasi waktu bermain pada *trial version* sebanyak 60 menit. Versi lengkap dari game dapat diperoleh dengan cara memasukkan sebuah kunci ketika game dimulai. Sebuah *keygen* dapat membuat kunci tersebut secara ilegal.
3. *Crack*
   Dengan cara ini, sebuah game dilepaskan dari sistem DRM-nya dengan menambahkan suatu file yang melakukan penyerangan terhadap DRM tersebut.

4. Teknik Anti Bajak untuk DRM pada Game

Untuk membangung sistem DRM anti pembajakan, dibutuhkan suatu algoritma enkripsi yang tidak dapat dipecahkan oleh teknik serangan apapun. Ada sebuah algoritma enkripsi sederhana namun sangat kuat dan tidak dapat terpecahkan dengan serangan kriptografik apapun (*unbreakable*), yang disebut sebagai algoritma enkripsi One Time Pad.
One Time Pad atau dikenal juga sebagai Vernam-cipher atau *perfect cipher* (*cipher* yang sempurna) adalah algoritma kriptografi yang menggabungkan plainteks dengan kunci random. Algoritma ini dikembangkan pada tahun 1917 oleh Gilbert Vernam untuk digunakan pada alat *telex*. Setiap kode Baudot 5 bit digabungkan dengan kode random sepanjang 5 bit juga pada selembar pita kertas. Pita-pita ini mengandung kode 5 bit yang banyak dan kemudian disebut sebagai *one-time-tape*. *One-time-tape* berjalan secara sinkron baik pada alat pengirim maupun pada alat penerima. Dari sinilah nama One Time Pad berasal, yaitu dari sejumlah lembar kertas di mana kunci random tersebut dicetak. One Time Pad dikenal sebagai algoritma enkripsi yang tahan serangan kriptografik.



Gambar 2 Mekanisme Enkripsi *One-time Tape*

4.1 Aturan One Time Pad

Ada sejumlah aturan yang harus diikuti agar One Time Pad tetap tahan serangan, jika salah satu aturan saja dilanggar, maka One Time Pad tidak dapat dikatakan tahan serangan. Aturan-aturan tersebut adalah sebagai berikut:

1. Panjang kunci sama dengan panjang plainteks
2. Kunci benar-benar random
3. Hanya boleh ada 2 salinan kunci, satu ada pada pengirim, dan satu lagi ada pada penerima
4. Kunci hanya boleh digunakan sekali saja, setelah digunakan, kunci harus segera dihancurkan

Aturan-aturan di ataslah yang menyebabkan algoritma ini *unbreakable*. Karena dengan ukuran kunci yang sama dengan plainteks dan isinya yang benar-benar random, pihak yang tidak memiliki kunci tidak mungkin melakukan observasi pola pemetaan plainteks ke cipherteks sehingga ia tidak mungkin berhasil melakukan dekripsi.

4.2 Cara Kerja One Time Pad

One Time Pad merupakan algoritma enkripsi file yang sederhana. Untuk melakukan enkripsi terhadap sebuah file A, maka dibutuhkan sebuah *one time pad* yang juga merupakan sebuah file dengan ukuran yang sama. Karakteristik file *one time pad* ini adalah sebagai berikut :

1. Ukuran file harus sama dengan file yang akan dienkripsi
2. File ini hanya boleh diketahui pengirim dan penerima pesan dan sebaiknya segera dihancurkan setelah pemakaian
3. File ini harus dibangkitkan dengan menggunakan PRNG (*pseudo-random number generator*)
4. Tidak boleh ada bagian yang dipakai berurutan pada file ini

Dapat dilihat bahwa karakteristik file *one time pad* ini harus memenuhi aturan One Time Pad yang telah dipaparkan di atas agar algoritma ini benar-benar tidak dapat dipecahkan.
Karena file *one time pad* harus benar-benar berisi kumpulan byte yang random, maka file *one time pad* harus dibangkitkan dengan menggunakan PRNG. Salah satu PRNG yang telah tersedia dan dapat langsung digunakan dengan mudah adalah sebuah kelas yang terdapat pada .NET Framework yang bernama RNGCryptoServiceProvider. Kelas ini mengimplementasi RNG (*Random Number Generator*) kriptografik dengan menggunakan implementasi pada CSP (*Cryptographic Service Provider*). Berikut adalah contoh implementasi fungsi pembangkit file *one time pad* dengan menggunakan kelas RNGCryptoServiceProvider.

```
using System;
using System.IO;
using System.Security.Cryptography;

class Onetimepad
{
    public bool Generate(string
strFilename, long nSize)
    {
        if (File.Exists(strFilename))
        {
            throw new
ArgumentException("File dengan nama
yang sama dengan file output ditemukan
");
        }

        FileStream theStream =
File.Create(strFilename);

        int nGenerateAtOnce = 1000;
        int nWriteNow =
nGenerateAtOnce;
        byte[] abStrongRBytes = new
Byte[nGenerateAtOnce];
        RNGCryptoServiceProvider rng =
new RNGCryptoServiceProvider();

        for (long nStart = 0; nStart
<= nSize; nStart += nGenerateAtOnce)
        {

rng.GetBytes(abStrongRBytes);
            if ((nStart +
nGenerateAtOnce) > nSize)
                nWriteNow =
Convert.ToInt32(nSize - nStart);

theStream.Write(abStrongRBytes, 0,
nWriteNow);
        }
        theStream.Close();
        return true;
    }
}
```

Lalu, bagaimana caranya mengenkripsi file dengan file *one time pad* ini? Caranya adalah dengan melakukan operasi XOR antara setiap byte pada file yang akan dienkripsi dengan sebuah byte pada file *one time pad*. Dengan cara ini, pihak yang tidak berkepentingan dengan file ini, tidak akan bisa melakukan dekripsi.Sedangkan pihak yang dimaksud, tidak akan mengalami kesulitan sedikitpun dalam melakukan dekripsi file, karena ia hanya tinggal melakukan enkripsi ulang terhadap file tersebut dan ia akan mendapatkan file aslinya. Proses enkripsi ulang ini dapat “membalikkan” file yang telah terenkripsi menjadi file aslinya karena operasi XOR memiliki aturan sebagai berikut :

plainteks = M

cipherteks = M ⊕ V

plainteks = cipherteks ⊕ V = ((M ⊕ V) ⊕ V)

Berikut adalah contoh implementasi fungsi yang melakukan proses enkripsi sebuah file dengan sebuah *one time pad*.

```
public long XorFileWithPad(string
strInputFile, string
strDestinationFile,
                                string
strPad, long nPadStartPos)
    {
        if (!File.Exists(strPad))
        {
            throw new
ArgumentException("File one time pad
tidak ditemukan");
        }

        if
(!File.Exists(strInputFile))
        {
            throw new
ArgumentException("File Input tidak
ditemukan");
        }

        if
(File.Exists(strDestinationFile))
        {
            throw new
ArgumentException("File dengan nama
yang sama dengan file output
ditemukan");
        }

        FileInfo infoPad = new
FileInfo(strPad);
        FileInfo infoInputFile = new
FileInfo(strInputFile);
        long nInputFileLength =
infoInputFile.Length;
        long nPadLength =
infoPad.Length;
        if ((nPadLength -
nPadStartPos) < nInputFileLength)
        {
            throw new
ArgumentException("Ukuran pad tidak
cukup untuk melakukan proses
enkripsi!");
        }

        FileStream fsOutput =
File.Create(strDestinationFile);
        FileStream fsPad =
File.OpenRead(strPad);
        FileStream fsInput =
File.OpenRead(strInputFile);
        int nBufferSize = 1000,
nInputSize, nPadSize, nXor;
        byte[] abInput = new
Byte[nBufferSize];
        byte[] abPad = new
Byte[nBufferSize];
        byte[] abOutput = new
Byte[nBufferSize];

        while (0 != (nInputSize =
fsInput.Read(abInput, 0,
nBufferSize)))
        {
            nPadSize =
fsPad.Read(abPad, 0, nBufferSize);
            for (nXor = 0; nXor <
nInputSize; nXor++)
                abOutput[nXor] =
Convert.ToByte(abInput[nXor] ^
abPad[nXor]);
            fsOutput.Write(abOutput,
0, nInputSize);
        }
        fsOutput.Close();
        fsInput.Close();
        fsPad.Close();

        // fungsi ini mengembalikan
byte terakhir yang digunakan pada one
time pad
        // jangan pernah menggunakan
ulang satu bagian pun pada one time
pad
        return (nPadStartPos +
nInputFileLength);
    }
```

4.3 Cara Kerja One Time Pad DRM

Seperti yang telah dijelaskan pada bagian sebelumnya, teknik kriptografi One Time Pad membutuhkan sebuah file *one time pad*. Ketika seorang *customer* membeli sebuah game, paket kiriman dari si produsen game berisi :

1. File game *trial version* yang telah terenkripsi dengan teknik One Time Pad sedemikian rupa sehingga tidak akan bisa dimainkan setelah batas waktu tertentu
2. Sebuah file *one time pad* yang berguna untuk mendekripsi file game *trial version* menjadi file game *full version*
3. Sebuah program yang melakukan proses enkripsi/dekripsi suatu file dengan file *one time pad* tertentu

Dengan demikian, karena sifat algoritma One Time Pad yang *unbrekable*, game *trial version* tidak akan dapat di-*crack* oleh siapapun tanpa menggunakan file *one time pad* yang mengenkripsinya. Namun, untuk menjaga keampuhan algoritma ini, semua aturan yang telah dikemukakan di atas harus dipenuhi.

5. Kesimpulan

Kesimpulan yang dapat diperoleh dari analisis DRM dan variasinya dan juga pencarian teknik DRM untuk game yang anti pembajakan adalah sebagai berikut :

1. DRM sangat penting untuk melindungi media digital dari praktik pembajakan
2. Teknik DRM yang digunakan untuk suatu game, tidak boleh melanggar hak orang yang telah secara legal mendapatkan game tersebut
3. Algoritma One Time Pad merupakan algoritma yang *unbreakable* karena adanya 4 aturan yang harus dipenuhi pada implementasinya, yaitu : panjang kunci harus sama dengan panjang plainteks, kunci benar-benar random, hanya boleh ada 1 salinan kunci masing-masing pada pengirim dan penerima, dan kunci hanya boleh dipakai sekali saja.
4. Game yang menggunakan sistem One Time Pad DRM tidak akan mungkin dibajak tanpa menggunakan file *one time pad* yang berasosiasi dengannya

DAFTAR PUSTAKA

[1] Arsenova, Emilija. (2005). *http://wob.iai.uni-bonn.de/Wob/images/01212504.pdf*. Tanggal akses : 11 Maret 2009 pukul 10.30.

[2] Bantick, Mike. (2008). *http://www.itwire.com/content/view/22159/1092/*. Tanggal akses : 11 Maret 2009 pukul 10.30.

[3] Carless, Simon. (2008). Opinion : 'Casual Games and Piracy: The Truth'. *http://www.gamesetwatch.com/2008/02/opopini_casual_games_and_pirac.php*. Tanggal akses : 11 Maret 2009.

[4] Nintendo. (2009). Nintendo Asks U.S. Trade Representative to Help Combat Global Video Game Piracy. *http://www.businesswire.com/news/home/20090225005461/en*. Tanggal akses : 11 Maret 2009 pukul 10.30.

[5] One-time Pad. (2009). *http://users.telenet.be/d.rijmenants/index.htt*. Tanggal akses : 1 April pukul 15.00.

[6] Wille, Christoph. (2001). *http://www.aspheute.com/*. Tanggal akses: 1 April 2009 pukul 15:00.