Pencegah Pembajakan Perangkat Lunak dengan Menggunakan Teknik Identity-Based Encryption dan Obfuscation

Wawan Wardiana
P2 Informatika-LIPI
wawan@informatika.lipi.go.id

Abstrak

Serangan-serangan terhadap proteksi hak cipta tentunya sangat merugikan pihak produsen yang telah susah payah membangun perangkat lunak ciptaannya. Teknik obfuscation *dapat digunakan untuk melindungi isi perangkat lunak dari pesaing atau penyerang yang ingin memanfaatkan sebagian/keseluruhan atau meniru/mengambil karya cipta digital seperti musik, gambar atau video untuk dimanfaatkan sebagai bagian produk lain dengan cara mengacak* source code *sehingga membuat sulit untuk dibaca dan dimengerti. Untuk melindungi perangkat lunak/aplikasi dari penggandaan, teknik* obfuscation *dapat dilengkapi dengan teknik enkripsi yang berbasis identitas (*Identity-Based Encryption/*IBE). Metodologi penelitian ini menggunakan metodologi pengembangan sistem perangkat lunak dengan menambahkan pembuktian/ metode formal terhadap aspek keamanannya, dengan menggunakan* Unified Modeling Language (*UML) sebagai visual modelling. Hasil uji coba penelitian ini cukup baik namun masih perlu dikembangkan dalam hal implementasi terhadap sistem di lapangan.*

Kata kunci: Copyright*, Pembajakan, IBE,* Obfuscation

1. Pendahuluan

Produk-produk berbasis teknologi informasi sekarang ini relatif mudah digandakan sehingga perlindungan terhadap produsen karya cipta digital menjadi penting. Salah satu bentuk perlindungan terhadap karya cipta digital adalah *Copyright*. Menurut undang-undang no 19/2002 *Copyright* (hak cipta) adalah hak eksklusif bagi Pencipta atau penerima hak untuk mengumumkan atau memperbanyak Ciptaannya atau memberikan izin untuk itu dengan tidak mengurangi pembatasan-pembatasan menurut peraturan perundang-undangan yang berlaku.

Perlindungan terhadap *copyright* untuk karya cipta digital (video, games, edutainment dan perangkat lunak) menjadi sangat penting jika dilihat dari dampak kerugian yang dirasakan oleh produsen dan negara dalam bentuk tergerusnya penjualan produk asli, hilangnya pendapatan pajak dan tidak tumbuhnya industri kreatif digital. Business Software Alliance pada tahun 2008 mengatakan bahwa pembajakkan pada produk digital di Indonesia pada tahun 2007 mencapai 84 %, [1]. Maraknya pembajakkan memang telah diperangi secara hukum dengan adanya UU no 19 tahun 2002 tentang Hak Cipta, namun hasil yang diharapkan belum optimal.

Produsen pada dasarnya dapat melindungi karya ciptanya sendiri. Sebagai produsen perangkat lunak, perlindungan hak cipta secara teknis dapat dilakukan dengan teknik-teknik yang digolongkan sebagai proteksi perangkat lunak. Proteksi perangkat lunak sesungguhnya juga merupakan produk yang dapat bersifat komersial mengingat banyak sekali karya cipta digital termasukkan industri rekaman film dan musik yang membutuhkan perlindungan dari penggandaan ilegal. Microsoft telah berinvestasi secara besar-besaran untuk membuat platform perangkat keras yang terpercaya. Lembaga-lembaga penelitian di Amerika Serikat mengeluarkan dana yang tidak sedikit, sekitar US$ 1.8 milyar untuk penelitian-penelitian tentang perlindungan perangkat lunak dengan *software* [1].

Masalah yang akan menjadi bahan kajian dan dicoba untuk diselesaikan adalah bagaimana melakukan pencegahan pembajakan yang sering dilakukan terhadap hasil karya digital berupa perangkat lunak, khususnya terhadap perangkat lunak yang dibuat dengan bahasa pemrograman untuk multimedia.

Tujuan penelitian ini adalah menerapkan sistem proteksi perangkat lunak pada karya cipta digital berupa perangkat ajar digital yang memiliki sifat off-line atau sekurang-kurangnya semi-online (interaksi antara *server* dan *client* hanya berlaku sekali), ekonomis sehingga tidak menaikkan harga jual secara signifikan dan tepat (perangkat lunak yang dijual masih dapat dijalankan pada lingkungan yang biasa), selain itu juga mempersulit pembajak untuk membaca *source code* yang sudah dibuat.

2. Study pustaka

2.1 *Identity based Encryption*

Salah satu pengembangan kriptografi yang paling menarik akhir-akhir ini adalah tentang *Identity Based Encryption* (IBE). Konsep dari IBE diperkenalkan oleh Shamir dalam papernya [2], bertujuan untuk memudahkan manajemen sertifikat dalam e-mail. Idenya adalah membuat sebuah kunci publik enkripsi dari serangkaian string yang berubah-ubah seperti alamat e-mail, nomor telepon, dsb.

Sejak tahun 1984 sampai dengan sekarang, sejumlah algoritma IBE telah muncul. Jon Callas dalam papernya [3] menganalisa beberapa metode IBE. Sistem original dari Shamir mengambil dasar dari enkripsi yang dibuat oleh Rivest, Shamir, dan Adlleman (RSA) dan merupakan sebuah sistem signature. Shamir masih belum memperluas metodenya sampai ke sistem enkripsinya. Clifford Cocks membuat sebuah skema berbasis residu kuadratik [4]. Sistem Cocks mengenkripsi bit-by-bit, dan membutuhkan ekspansi dari pesan. Boneh dan Matt Franklin membuat sebuah skema berbasis Weil Pairings [5]. Pairing menggunakan pemetaan bilinear antar kelompok untuk menentukan sebuah relationship dimana pemrosesan identitas dengan fungsi hash digunakan untuk membuat sebuah sistem enkripsi. Boneh-Franklin IBE (BF-IBE) telah digunakan sampai sekarang [6] dan masih menjadi wilayah penelitian untuk pengembangan lebih lanjut. Horwitz and Lynn [7], Gentry and Silverberg [8] mengembangkan karakteristik kemampuan dari BF-IBE Public Key Generator (PKG) dengan mengembangkan sistem IBE system menjadi Hierarchical IBE (HIBE).

Apa yang mereka kerjakan merupakan sesuatu yang penting karena fokus dari metode ini adalah untuk mempraktekkan detail dari pembuatan sebuah PKG yang terukur. Gentry juga mendeskripsikan *Certificate-Based Encryption* (CBE) yang menggunakan sebuah sistem IBE dengan sertifikat untuk membuat sebuah pendekatan *hybrid* [8], yang secara esensial membuat "identitas" bukan cuma sebuah nama, tetapi sebuah sertifikat yang terdefinisi dengan baik. Dalam sebuah konsep pendekatan, Al-Riyami and Paterson mempunyai *Certificateless Public Key Cryptography* [9] Benoît Libert and Jean-Jacques Quisquater juga membuat sebuah skema *identity-based signcryption* berbasis *pairing* [10]. Skema ini mengkombinasikan keseluruhan aspek dalam satu operasi., dan juga ada beberapa metode lain yang berhubungan dengan pengkombinasian signing dan enkripsi seperti yang diutarakan oleh Zheng [11].

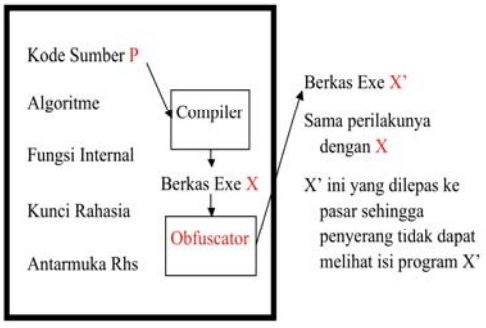
2.2 Teknik obfuscation

Obfuscation atau obfuskasi adalah mentransformasi sintaks kode komputer namun dengan tetap memelihara semantik (isi) sehingga kerahasiaan tetap terjaga [12]. Berapa aspek yang mestinya tetap menjadi rahasia dalam sebuah aplikasi komputer yaitu, [13]:

a. Algoritma, sehingga kompetitor tidak dapat membangun hal yang sama kecuali membangunnya dari awal.
b. Konstanta, seperti kunci enkripsi

c. Fungsi internal yang penting, seperti fungsi untuk mengecek lisensi "if (not licensed) exit()"
d. Antarmuka eksternal, untuk menolak akses dari penyerang dan competitor sehingga dapat masukan dari "pintu belakang" atau "lubang".

Pada gambar 1 ditunjukkan skenario obfuskasi pada sebuah aplikasi. Sebuah aplikasi yang di dalamnya mengandung algoritma, fungsi internal, kunci dan antarmuka rahasia kemudian dikompilasi sehingga menghasilkan berkas yang dapat dieksekusi X. Namun berkas yang dapat dieksekusi ini bisa dengan mudah dilihat isinya oleh penyerang (pembajak). Oleh karena itu dibuat sebuah sistem yang disebut *obfuscator* yang gunanya mengacaukan sintaks berkas yang dapat dieksekusi. Berkas yang dihasilkan oleh o*bfuscator* disebut X' masih dapat dieksekusi namun dengan sintaks kode yang terkacaukan. Perilaku X' sama persis dengan X dengan obfuskasi isi berkas terlindungi.



Gambar 1. Skenario Obfuscation [13]

Cara-cara untuk meng-obfuskasi perangkat lunak dapat digolongkan menjadi:

a. Obfuskasi Leksikal: kacaukan nama variabel, konstant, metode, kelas-kelas, antarmuka dsb.
b. Obfuskasi Data: kacaukan nilai variabel (misalnya dengan menkode boolean menjadi int, mengkode int pada flat; menkode nilai-nilai pada graph)
c. Obfuskasi Kendali: kacaukan pernyataan-pernyataan kendali (if,while,for).

Teknik obfuskasi dapat digunakan untuk melindungi isi perangkat lunak dari pesaing atau penyerang yang ingin memanfaatkan sebagian/ keseluruhan atau meniru/ mengambil karya cipta digital seperti musik, gambar atau video untuk dimanfaatkan sebagai bagian produk lain. Untuk melindungi perangkat lunak/aplikasi dari penggandaan teknik obfuskasi dapat dilengkapi dengan teknik enkripsi yang sudah dijelaskan sebelumnya di atas..

3. Metode

a. Metodologi penelitian ini menggunakan metodologi pengembangan sistem perangkat lunak dengan menambahkan pembuktian/ metode formal terhadap aspek keamanannya. Berikut ini adalah langkah-langkah yang ada pada metodologi penelitian ini:
b. Analisis Kebutuhan, pada tahap ini merupakan cara untuk mengungkapkan kebutuhan sistem proteksi perangkat lunak.
c. Merancang Skema Proteksi, berdasarkan hasil analisis kebutuhan, skema proteksi terhadap penggandaan dan pencurian kode dibuat dengan berbasiskan 2 teknik yaitu obfuscation dan identity based-encryption.
d. Membuktikan Skema Proteksi Secara Formal, skema proteksi yang dibuat dibuktikan dengan menggunakan metode formal. Metode formal adalah cara deduksi yang menggunakan perangkat logika matematika untuk menurunkan apakah sebuah skema adalah sesuai dengan asumsi atau tidak (aman atau tidak). Metode formal untuk keamanan bias menggunakan piranti-piranti yang tersedia secara open source seperti SPIN.
e. Implementasi Sistem, pada tahap implementasi sistem, sistem proteksi perangkat lunak dirancang, dibuat dan diuji secara modular dengan menggunakan prinsip-prinsip rekayasa perangkat lunak.

f. Memvalidasi dan memverifikasi Sistem, sistem yang dibuat pada tahap 4 diverifikasi berdasarkan permintaan yang dibuat pada tahap 1. Dalam tahap ini diverifikasi juga tentang aspek ekonomis selain aspek teknis. Validasi bertujuan untuk menguji ketepatan sistem dan verifikasi bertujuan untuk menguji apakah sistem sesuai dengan permintaan.
g. Memaket dan Integrasi Sistem, setelah sistem tervalidasi dan terverifikasi, sistem proteksi dipaketkan dan diintegrasikan pada sistem produksi yang sudah ada.

4. Hasil penelitian dan pembahasan

Teknik obfuskasi dan enkripsi yang berbasis identitas digunakan untuk melindungi produk perangkat lunak (berupa CD multimedia). Teknik obfuskasi digunakan untuk mengacak skrip dan referens ke berkas sehingga isi berkas (game, video, gambar, lagu) tidak dapat dibuka/dibongkar oleh pihak yang tidak bertanggung jawab.

Pada penelitian ini teknik obfuskasi diterapkan pada skrip flash bertujuan untuk menyulitkan seseorang untuk mendeduksi isi skrip sehingga ia tidak dapat membaca skrip untuk mengetahui logika skrip (untuk games) atau dimana file dan kunci penting.

Pada intinya obfuskasi digunakan untuk mencegah adanya *reverse engineering* dari sebuah perangkat lunak. Teknik obfuskasi umumnya mengubah sintaks skrip tanpa mengubah semantiknya., sehingga walaupun tulisan skrip menjadi susah untuk dibaca namun ketika dieksekusi masih tetap dapat dijalankan seperti sebelum di obfuskasi. Contoh skrip program sebelum obfuskasi dan sesudah proses obfuskasi ditunjukkan pada gambar 2 dan gambar 3.

Skrip hasil obfuskasi akan sulit dibaca meskipun bila dijalankan pada kompiler yang sama akan menghasil keluaran yang sama. Obfuskasi dapat diterapkan di objek hasil compile dalam hal ini objek-objek komponen flash. Namun untuk menerapkan obfuskasi pada objek-objek flash diperlukan interpreter yang standard dan harus dimodifikasi agar dapat menjalankan objek flash yang terobfuskasi. Secara konsep teknik obfuskasi pada objek dilakukan dengan menambahkan interpreter flash dengan tambahan pustaka sehingga intepreter itu dapat menjalankan objek flash yang sudah di obfuskasi. Skema obfuskasi pada objek seperti digambarkan pada gambar 4. Sedangkan hasil implementasi antarmukanya seperti terlihat pada gambar 5.

```
void primes(int cap) {
  int i, j, composite;
  for(i = 2; i < cap; i++) {
    composite = 0;
    for(j = 2; j < i; j++)
      composite += !(i % j);
    if(!composite)
      printf("%d\t", i);
  }
}
 int main() {
  primes(100);
}
```

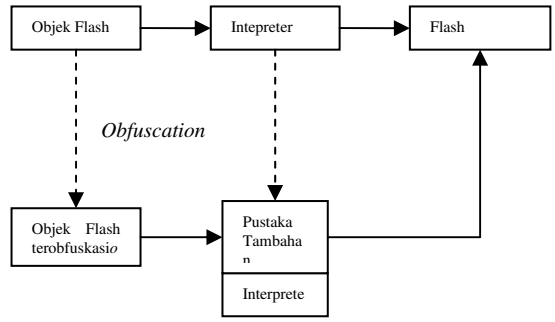
Gambar 2. Contoh skrip normal

```
_(__,___,____){___/__<=1?_(__,___+1,
____):!(___%__)?_(__,___+1,0):___%__
==___/

__&&!____?(printf("%d\t",___/__),_(_
_,___+1,0)):___%__>1&&___%__<___/__?
_(__,1+

___,____+!(___/__%(___%__))):___<__*
__?_(__,___+1,____):0;}main(){_(100,
0,0);}
```

Gambar 3. Contoh skrip sesudah diobfuskasi

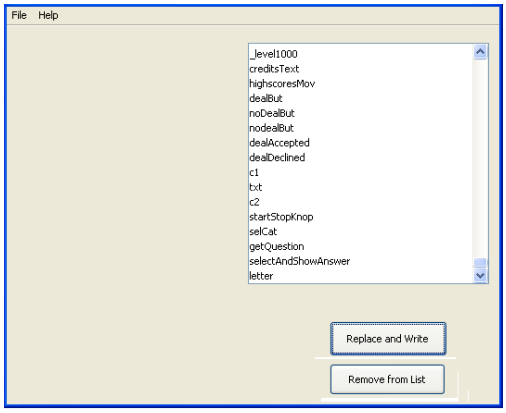


Gambar 4. Skema obfuskasi pada objek

Teknik kedua yang dipakai adalah dengan enkripsi berbasis ID. Enkripsi berbasis ID ini digunakan untuk mengotentikasi pengguna resmi. Jika pengguna memiliki kunci *private* yang tepat maka pengguna dapat menginstall aplikasi

sehingga dapat dijalankan. Pada dasarnya skema otentikasi yang dipakai sama dengan skema aktivasi yang umumnya dipakai yaitu pengguna diberikan kode aktivasi sehingga aplikasi dapat digunakan secara penuh, hanya saja dalam penelitian ini kode sumbernya sudah di acak / *obfuscated file*.

Desain dari metode ini dapat dilihat sebagai modifikasi dari skema DES dengan beberapa penambahan pada *key generator* dan penggunaan identitas pengguna sebagai kunci publik enkripsi. [9] [10]. Gambar 6 menunjukkan proses otentikasi perangkat lunak menggunakan algoritma IB-DES. Pengguna mengirimkan identitas mereka (misalkan nama, alamat e-mail, dan nomor telepon) serta *serial number* produk ke *server* (dalam hal ini misalkan *software* distributor/penjual perangkat lunak).



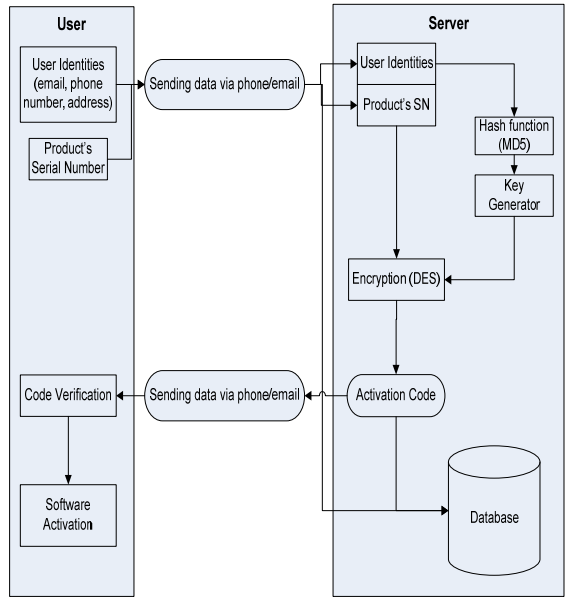
Gambar 5. Interfacing area kerja obfuskator

Kemudian server akan memproses identitas ini untuk membangkitkan kode aktivasi. Pertama, fungsi hash akan mengubah identitas pengguna menjadi digest 128-bit dan menggunakannya sebagai kunci enkripsi. Kedua, serial number produk akan dienkripsi oleh algoritma DES menggunakan kunci yang dihasilkan oleh key generator.

Kode aktivasi merupakan hasil dari proses enkripsi ini dan akan dikirim ke pengguna untuk mengaktivasi perangkat lunaknya. Baik identitas maupun kode aktivasi akan disimpan dalam database server. Langkah ini diperlukan untuk mencegah duplikasi perangkat lunak secara ilegal oleh pengguna.

Detail dari algoritma IB-DES diilustrasikan pada gambar 7 di bawah ini. Masukan dari fungsi MD5 adalah identitas pengguna. Sedangkan masukan untuk enkripsi DES adalah serial number produk perangkat lunak dan hasil dari fungsi hash yang digunakan oleh key generator. Keseluruhan proses dibagi menjadi 3 bagian: bagian 1 adalah skema DES dengan n round, bagian 2 adalah key transformation, dan bagian 3 adalah key generator untuk n round. Key transformation digunakan untuk mentransformasi kunci dari 128-bit menjadi 56-bit. Key generator digunakan untuk membuat sub kunci yang diperlukan oleh skema DES pada tiap perulangannya.

Kemudian, *server* akan memproses identitas ini untuk membangkitkan sebuah kode aktivasi. Pertama, fungsi hash akan mengubah identitas pengguna menjadi digest 128-bit dan menggunakannya sebagai kunci.



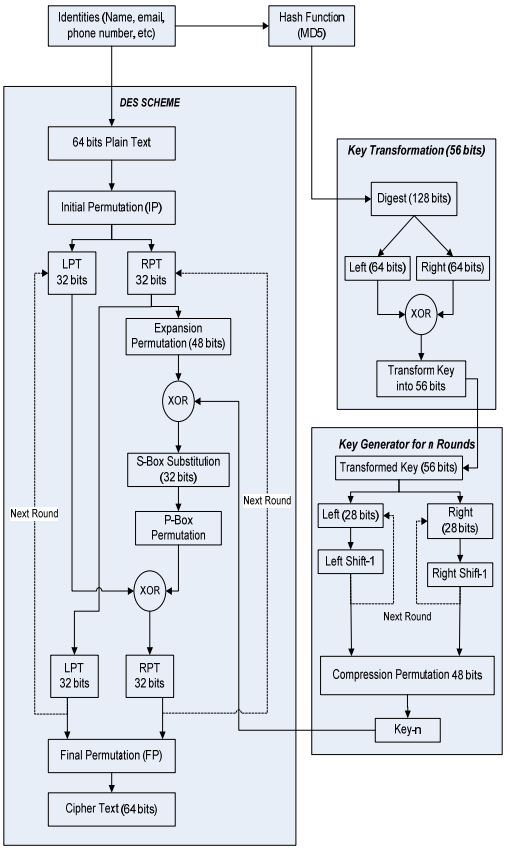
Gambar 6. Otentikasi Perangkat Lunak Menggunakan IB-DES

4.1 Skema DES

Pada langkah pertama, blok plainteks 64 bit ditangani oleh fungsi Initial Permutation (IP). IP hanya dilakukan sekali, dan dilakukan sebelum perulangan pertama. Fungsi permutasi adalah transposisi bit

berdasarkan tabel yang telah kita definisikan sebelumnya, misalkan IP menempatkan bit pertama blok pada bit ke-58 blok, bit ke-2 pada bit ke-50, dst. Selanjutnya, keluaran dari IP dibagi menjadi dua bagian, Left Plain Text(LPT) dan Right Plain Text(RPT). Sekarang, masing-masing LPT dan RPT akan diproses melalui n round proses enkripsi dengan menggunakan sub kuncinya masing-masing.

Terakhir, LPT dan RPT digabungkan lagi dan sebuah fungsi Final Permutation (FP) digunakan untuk mengkombinasikan blok ini. FP merupakan invers dari IP. Hasil dari proses di atas menghasilkan ciphertext 64 bit.



Gambar 7. Skema IB-DES

4.2 Tranformasi kunci

Tujuan utama dari key transformation adalah untuk mentransformasi 128 bit kunci menjadi 56 bit kunci. Langkah-langkanya yaitu: Pertama, 128-bit kunci dibagi menjadi dua bagian, 64-bit Left dan 64-bit Right. Kemudian, Left dan Right digabungkan dengan menggunakan operasi XOR. Proses ini menghasilkan 64 bit kunci. Kunci 64 bit ini kemudian ditransformasi dengan cara menghapus bit ke-8 dan kelipatannya. Dari kunci 56 bit ini, sub kunci 48 bit yang berbeda-beda akan diproduksi oleh key generator.

4.3 Otentikasi

Konsep dari otentikasi perangkat lunak adalah untuk memverifikasi kode aktivasi yang dimasukkan pengguna dengan kode yang diproses oleh perangkat lunak, jika kode tersebut sama maka perangkat lunak akan teraktivasi. Langkah-langkah pemrosesan kode dalam produk perangkat lunak mirip dengan pemrosesan kode pada sisi server.

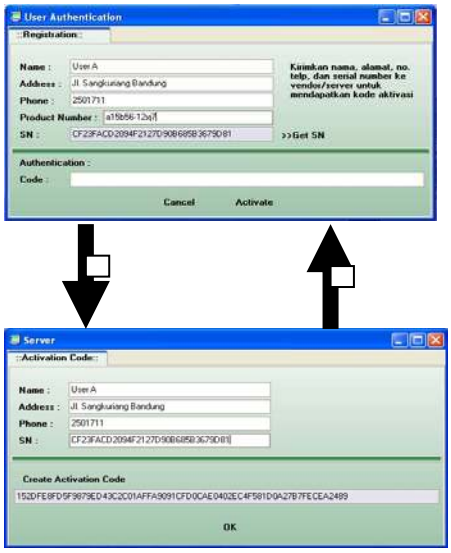
Perbedaannya hanya pada penyimpanan beberapa variable dalam sebuah database yang dilakukan hanya di server. Server membutuhkan penyimpanan data-data tersebut (identitas pengguna, serial number produk, kode aktivasi) dalam database adalah untuk mencegah duplikasi produk secara ilegal. Hal ini dapat diketahui dengan mengecek data pengguna, yaitu jika ada data pengguna dengan identitas dan nomor seri produk yang sama. Sedangkan produk software tidak memerlukan database untuk menyimpan data otentikasi karena pada sisi ini sistem hanya melakukan proses otentikasi saja.

4.4 Aplikasi IBE-DES

Aplikasi IBE-DES merupakan sebuah software yang digunakan untuk meregistrasi produk perangkat lunak dengan tujuan mencegah pembajakan perangkat lunak tersebut. *Software* ini mempunyai 2 paket sistem, pertama adalah paket registrasi user yang harus digabung/ditempelkan pada produk perangkat lunak, dan kedua adalah paket pembuat kode aktivasi di *server*. Paket kedua ini digunakan sekaligus untuk manajemen distribusi produk dengan mencatat identitas *user*, *serial number user*,

dan kode aktivasinya, sehingga masing-masing user hanya bisa menggunakan produk sesuai dengan lisensinya.

Contoh tampilan antarmuka hasil implementasi masing-masing paket sistem tersebut ditunjukkan oleh gambar 8 di bawah ini. Ilustrasi proses dari gambar tersebut adalah sebagai berikut; ketika user menggunakan produk pertama kali, maka software akan menanyakan identitas user dan nomor produk yang selanjutnya digunakan untuk membangkitkan *serial number* dari produk tersebut. Serial Number dibangkitkan dari proses enkripsi CPU-ID (nomor unik komputer user) dan nomor produk sehingga diperoleh serangkaian kode. Untuk bisa menjalankan produk, *user* harus meminta kode aktivasi dari *server* dengan mengirimkan identitas yang telah ditulis di awal beserta serial number ke *server*. Setelah *server* mengirimkan kode aktivasi, sistem di user akan melakukan otentikasi kode. Jika kode yang diberikan sesuai, maka produk telah diregistrasi dan siap untuk digunakan.



Gambar 8. Antarmuka proses otentikasi menggunakan IBE-DES

Dengan penambahan kedua paket sistem di atas, diharapkan bisa mencegah pembajakan perangkat lunak dan juga mempermudah manajemen pendistribusian perangkat lunak. Hal ini berarti juga menaikkan perekonomian dan perlindungan terhadap hak cipta.

5. Kesimpulan dan saran

5.1 Kesimpulan

Proteksi perangkat lunak secara teoritis merupakan gabungan antara ilmu keamanan komputer, kriptografi, rekayasa perangkat lunak dan rekayasa perangkat keras. Namun dalam kenyataannya membuat sistem pelindung perangkat lunak bukanlah hal yang mudah karena dalam pembuatan skema perlindungan biasanya digunakan asumsi-asumsi menyangkut kondisi perangkat keras dan lingkungan.

Teknik *code obfuscation* bisa diterapkan pada produk perangkat lunak yang bersifat aplikasi namun tidak dapat diterapkan pada karya cipta digital. Teknik yang dibuat pada penelitian ini baru diimplementasikan pada perangkat lunak yang dibuat dengan program macromedia dengan ekstensi SWF.

Identity Based Encryption menyediakan sebuah paradigma sederhana untuk mengimplementasikan kunci publik. Metode ini membuat pengguna akhir mudah untuk memverifikasi perangkat lunak yang mereka beli. Pengguna hanya perlu mengirimkan identitas mereka dan nomor seri produk dan mereka akan mendapatkan kode aktivasi. Skema *Data Encryption Standard* (DES) digunakan untuk melindungi proses otentikasi dan memproduksi kode aktivasi. Sebuah skema untuk mencegah pembajakan perangkat lunak masih diperlukan dan masih merupakan sebuah problem terbuka.

Penelitian ini menggabungkan kedua teknik di atas. Uji coba sudah dilakukan namun belum diimplementasikan pada lingkungan yang sesungguhnya. Mungkin saja ada perbedaan yang signifikan terhadap perilaku sistem pada lingkungan yang berbeda, baik lingkungan perangkat lunak pengguna maupun perangkat lunak yang akan dilakukan proteksinya.

5.2 Saran

a. Perlu lebih disempurnakan lagi tentang fungsi perangkat lunak ini dalam

implementasi di lingkungan yang sesungguhnya, sehingga kegiatan ini perlu dilanjutkan pada tahap berikutnya.

b. Perlunya dipikirkan langkah selanjutnya dalam pengembangan *packaging* perangkat lunak dalam bentuk yang siap dipasarkan, dengan kemasan yang menarik.

6. Ucapan terimakasih

Pada kesempatan ini kami mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada Kementerian Negara Riset dan Teknologi yang telah membiayai penelitian ini. Kepada rekan-rekan peneliti Ana Heryana, Novahadi Lestriandoko, Rifki Sadikin, yang berada di Pusat Penelitian Informatika-LIPI yang telah membantu dalam persiapan dan pelaksanaan penelitian ini.

7. Daftar pustaka

[1] BSA, Illegal PC Software use down to 84% in 2007 in Indonesia, http://w3.bsa.org/indonesia/press/newsreleases/globalstudypr.cfm, akses terakhir 19 Agustus 2008

[2] Shamir (1985), Identity-Based Cryptosystems and Signature Schemes, Proceedings of CRYPTO'84, LNCS 196, pages 47-53.

[3] Callas (2005), Identity-Based Encryption with Conventional Public-Key Infrastructure, Proceedings of the 4th Annual PKI R&D Workshop: Multiple Paths to Trust, NIST, Gaithersburg MD.

[4] C. Cocks (2001), An Identity Based Encryption Scheme Based on Quadratic Residues, Proceedings of the 8th IMA International Conference on Cryptography and Coding, LNCS 2260, pages 360-363, Springer-Verlag.

[5] D. Boneh and M. Franklin (2001), Identity-Based Encryption from the Weil Pairing, Proceedings of CRYPTO'01, LNCS 2139, pages 213-229, Springer-Verlag.

[6] D. Boneh and X. Boyen (2004), Secure Identity Based Encryption Without Random Oracles, extended abstract in Proceedings of CRYPTO'04, LNCS 3152, Springr-Verlag, Full paper is available in the IARC eprint archives, http://eprint.iacr.org/2004/173/.

[7] J. Horwitz and B. Lynn (2002), Toward Hierarchical Identity-Based Encryption, Proceedings of EUROCRYPT'02, LNCS 2332, pages 466-481, Springer-Verlag.

[8] C. Gentry and A. Silverberg (2002), Hierarchical ID-Based Cryptography, Proceedings of ASIACRYPT'02, LNCA 2501, pages 548-566, Springer-Verlag. Corrected version available as http://eprint.iacr.org/2002/056/.

[9] S. Al-Riyami and K. G. Paterson (2003), Certificateless Public Key Cryptography, extended abstract in Proceedings of ASIACRYPT '03, LNCS 2894, Springer-Verlag. Full paper is available in the IARC eprint archives, http://eprint.iacr.org/2003/126/

[10] B. Libert and J. Quisquater (2003), New Identity Based Signcryption Schemes from Pairings, IEEE Information Theory Workshop, Also available as http://eprint.iacr.org/2003/023/.

[11] Y. Zheng (1997), Digital Signcryption or to achieve cost (signature & encryption) << cost(signature) + cost(encryption), Proceedings of CRYPTO'97, LNCS 1294, pages 165-179, Springer-Verlag.

[12] Coliberg CS. (2002), Watermarking, Temper-Proofing and Obfuscation – Tools for Software Protection, IEEE Transc On Software Engineering Vol 28 No 6

[13] Clark Thomborson. (2007), Methods for Software Protection, Keynote Address on International Forum on Computer Science and Advanced Software Technology, Jianxi Normal University.