PENERAPAN ALGORITMA KRIPTOGRAFI KUNCI PUBLIK SEBAGAI PENGAMANAN SISTEM DISTRIBUSI PERANGKAT LUNAK LIPIRISM@

Wawan Wardiana[1)], Ana Heryana[2)]
Pusat Penelitian Informatika - LIPI
Jl. Sangkuriang / Cisitu 21/154D Bandung 40135 Telepon (022) 2504711
E-mail : wawan@informatika.lipi.go.id[1)], aheryana@informatika.lipi.go.id[2)]

Abstrak

LIPIRISm@ adalah sebuah perangkat lunak yang telah dibuat oleh Pusat Penelitian Informatika – LIPI yang digunakan oleh para praktisi iridologi untuk menganalisa kondisi organ dan sistem tubuh melalui iris mata seseorang. Pada tahun ini perangkat lunak LIPIRISm@ akan diluncurkan, namun tentunya untuk menjaga keaslian dan distribusi perangkat lunak ini maka diperlukan suatu sistem pengamanan yang baik agar tujuan dibuatnya perangkat lunak ini dapat terpenuhi. Pada tulisan ini akan dibahas beberapa model pengamanan perangkat lunak, dan yang dipilih adalah model pengamanan Sistem Registrasi Online dengan menerapkan Algoritma Kriptografi Kunci Publik pada perangkat lunak LIPIRISm@ dengan berbagai argumentasinya.

Kata Kunci : Perangkat Lunak LIPIRISm@, Sistem Pengamanan, Sistem Registrasi Online.

PENDAHULUAN

Pusat Penelitian Informatika sebagai salah satu lembaga pemerintah di bawah Lembaga Ilmu Pengetahuan Indonesia telah menghasilkan perangkat lunak yang dapat digunakan oleh para praktisi iridologi untuk membantunya pada saat melakukan analisa / diagnosa kondisi organ dan sistem tubuh seseorang. Perangka lunak ini dinamakan *LIPIRISm@.*

LIPIRISm@ dibuat dengan tujuan agar para praktisi iridologi di Indonesia khususnya dapat melakukan analisa dan diagnosanya dengan lebih mudah dan lebih akurat, karena pada perangkat lunak ini diimplementasikannya suatu peta digital *(digital chart)* yang biasa digunakan oleh para praktisi yakni peta iris mata yang dibuat oleh Dr. Bernard Jensen. Diluar negeri sendiri perangkat lunak seperti ini sudah ada beberapa yang beredar namun harganya masih relatif mahal.

Seringkali kita sebagai produsen atau pengembang perangkat lunak untuk suatu sistem atau aplikasi dibuat terheran-heran bila melihat penyebaran atau distribusi perangkat lunak di Indonesia, saat produsen baru mengeluarkan secara resmi versi beta tetapi kenyataan dilapangan masyarakat sudah memiliki versi yang lengkap dengan harga yang sangat murah, dan yang paling menyakitkan lagi adalah bahwa perangkat lunak yang didapatkan oleh masyarakat adalah perangkat lunak bajakan. Bukan perangkat lunak asli yang dibuat dan didistribusikan oleh pihak produsen atau pihak ketiga dengan sepengetahuan produsen.

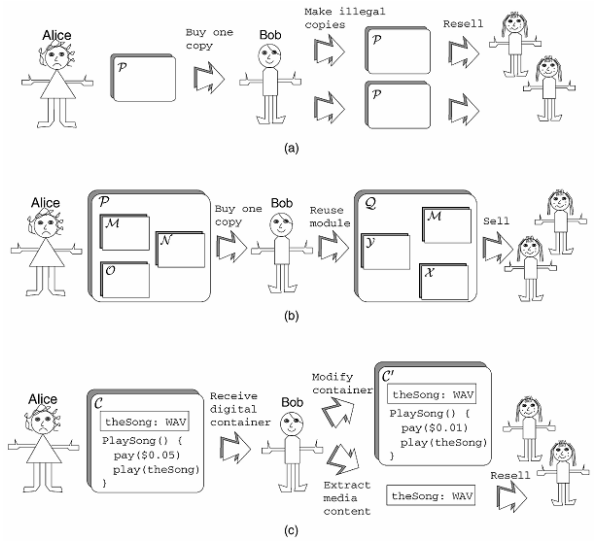
Sekarang ini pembajakan di Indonesia begitu maraknya, bahkan *Business Software Aliance* (BSA) mengatakan pembajakan perangkat lunak di Indonesia pada tahun 2007 mencapai 84% [1], bisa dibayangkan berapa uang yang seharusnya diperoleh oleh para produsen perangkat lunak ini seandainya tidak terjadi pembajakan.

Masalah pembajakan perangkat lunak masih tetap menghantui siapa saja, dan menjadi ancaman serius bagi para pengembang perangkat lunak melalui *software house* yang sekarang menjadi tren penting perkembangan teknologi informasi dan komunikasi. Melindungi hasil karya pembuatan perangkat lunak memang menjadi sebuah isu penting di tengah-tengah tingginya angka pembajakan.

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 [2].

Bentuk-bentuk pembajakan dapat dilihat pada ilustrasi seperti pada gambar 1 di bawah ini. Seseorang dapat membeli perangkat lunak dengan copy yang resmi, tetapi kemudian dia melakukan penyalinan ulang dan menjualnya dengan harga yang lebih murah. Bentuk lainnya adalah dengan membeli salinan resmi kemudian menggunakan modul-modul yang ada pada salinan tersebut untuk kemudian digabungkan dengan modul-modul lainnya yang dia miliki untuk dijual kembali dengan harga yang relatif lebih murah. Dan bentuk lainnya adalah dengan menggunakan karya

cipta yang ada dalam suatu perangkat lunak seperti lagu, film, ataupun gambar untuk digunakan oleh pihak yang tidak berhak.



Gambar 1. Bentuk pembajakan yang sering dilakukan oleh masyarakat [3]

Untuk menjaga keaslian dan distribusi perangkat lunak *LIPIRISm@* ini, maka diperlukan suatu model pengamanan, agar perangkat lunak digunakan oleh orang yang memang berhak dan dengan cara pemilikan yang sah. Perlindungan terhadap karya perangkat lunak memang menjadi sebuah keharusan, bukan hanya untuk melindungi bisnis perangkatnya sendiri, tetapi juga melindungi bisnis keseluruhan, termasuk bisnis pengguna perangkat lunak itu sendiri yang memberikan kontribusi pada pertumbuhan ekonomi.

METODOLOGI PENELITIAN

Metode-metode pengamanan perangkat lunak sekarang ini sangatlah beragam, baik cara maupun teknik yang digunakannya.

Dalam penelitian ini akan dilakukan beberapa kegiatan yang pada akhirnya akan menghasilkan suatu model sistem pengamanan yang sesuai dengan karakteristik perangkat lunak maupun penggunanya. Diawali dengan pengumpulan data tentang karakteristik dari perangkat lunak *LIPIRISm@* dan para penggunanya, kemudian akan diinventarisir teknik-teknik pengamanan perangkat lunak yang kemudian akan dipelajari dan dibandingkan antara satu dengan yang lainnya, sehingga akan didapatkan suatu teknik pengamanan perangkat lunak *LIPIRISm@* yang sesuai.

Kemudian akan dilakukan perancangan terhadap sistem pengamanan distribusi perangkat lunak *LIPIRISm@* yang sesuai dengan teknik pengamanan yang sudah dipilih sebelumnya yakni dengan menggunakan algoritma kriptografi kunci publik. Dan kegiatan terakhir adalah mengimplementasikannya serta menguji kehandalan dari rancangan pengamanan yang sudah dibuat sebelumnya.

HASIL DAN PERANCANGAN

Pada bab ini akan dibahas mengenai beberapa model pengamanan perangkat lunak dengan keterbatasan dan kelebihan masing-masing model.

Teknik-teknik proteksi perangkat lunak terus menerus berkembang, namun sebaiknya perangkat lunak proteksi yang dibuat harus memenuhi kriteria-kriteria agar dianggap layak. Martin dan Hughes [4] menyebutkan beberapa kriteria yakni :

- *Availability* : Untuk user yang berhak dapat mengakses data dan informasi tepat pada waktunya
- *Authentication* : adanya jaminan terhadap validasi pengirim, pesan yang dikirim, bentuk transmisi, dan verifikasi terhadap penerimanya
- *Confidentiality* : Adanya jaminan yang membaca adalah yang berhak dan telah diberikan otoritas.
- *Integrity* : Adanya penggabungan antara perangkat lunak dan perangkat keras dalam proteksi harus bisa menjamin konsistensi terhadap struktur data dan perubahannya dalam media penyimpanan data.
- *Non-repudiation :* Adanya jaminan persetujuan antara pengirim dan penerima data, sehingga tidak akan adanya penolakan pada saat data diproses.

Berikut adalah beberapa teknik yang bisa dilakukan untuk memproteksi perangkat lunak dari usaha pembajakan. Menurut Thomborson [5] :

- Dengan menggunakan sebuah *dongle*, *dongle* adalah sebuah piranti keras yang mengandung angka serial yang harus dipasang dikomputer agar perangkat lunak dapat dijalankan. Penambahan ini menambah biaya biasanya *dongle* dipakai pada perangkat lunak yang berbiaya mahal, jarang ditemukan *games* atau *edutaiment* yang menggunakan *dongle*. Sekarang beberapa perusahaan perangkat lunak sudah mulai mengembangkannya dengan *USB Dongle.*
- Enkripsi *Bus,* bus adalah jalur elektronik yang digunakan untuk interaksi media penyimpanan ke prosessor. Namun pilihan pada solusi proteksi ini masih bersifat perangkat keras dan biasanya diterapkan pada mesin ATM.
- Kunci registrasi, cara ini adalah yang umum digunakan dan murah. Kunci registrasi merupakan serial angka/huruf yang diminta ketika akan menjalankan program. Banyak piranti games/edutaiment menggunakan kunci registrasi.
- Nama dan serial. Nama dan angka serial digunakan untuk memproteksi perangkat lunak. Sama seperti kunci registrasi bila nama dan serial bocor maka dapat dengan mudah pihak tak bertanggungjawab menggandakan perangkat lunak.
- Aktivasi produk melalui internet. Cara ini sebenarnya ektensi dari cara nama dan serial

namun pemberian serial melalui internet dan tercatat di server pusat. Cara ini banyak digunakan oleh perangkat lunak, games dan edutainment namun mengasumsikan adanya infrastruktur internet pada konsumen.

- Proteksi dengan menggunakan *code obfuscation*. *Code obfuscation* mengubah kode biner menjadi kode-kode yang terlihat berantakkan sehingga tidak dapat dijalakan dengan kompiler biasa.

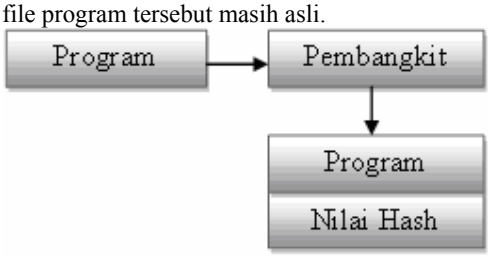
SEGMEN PASAR PERANGKAT LUNAK

Perangkat lunak *LIPIRISm@* adalah perangkat lunak *special purpose* sehingga penggunanyapun bukan pengguna awam, tetapi pengguna yang memiliki latar belakang kelimuan iridologi. Namun karena banyaknya para praktisi di bidang iridologi ini, maka sudah selayaknya pula *LIPIRISm@* ini diproteksi dalam distribusinya untuk menjaga keaslian perangkat lunak ini dari perubahan oihak luar, juga untuk membatasi penggunaan perangkat lunak ini hanya oleh yang mempunyai haknya saja yang sudah membeli secara legal.

Dengan melihat teknik dan metode dalam proteksi perangkat lunak di atas, kemudian diselaraskan dengan sistem distribusi perangkat lunak *LIPIRISm@*, maka dipilihlah teknik pengamanan dengan metoda kunci registrasi dan melakukan aktivasinya melalui internet. Untuk itu dilakukan perancangan dengan menggunakan algoritma kriptografi kunci publik.

ALGORITMA KUNCI PUBLIK

Menurut Kusumah [6] Salah satu metode yang dapat digunakan untuk memeriksa keaslian file program adalah dengan menghitung nilai *hash* dari file tersebut dan menyimpan pada tempat tertentu, misalnya ditambahkan (*embed*) di akhir file asli seperti terlihat pada gambar 2 di bawah ini. Kemudian ketika program berjalan, di awal proses ia menghitung kembali nilai *hash* untuk dirinya sendiri, dan membandingkannya dengan nilai *hash* yang telah disimpan sebelumnya. Bila kedua nilai sama, maka dapat diasumsikan bahwa file program tersebut masih asli.

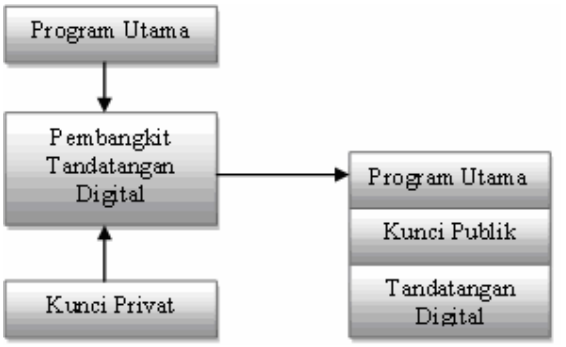


Gambar 2. Skema proteksi dengan menggunakan fungsi *hash*

Kelemahan metode ini adalah apabila pihak luar ingin memodifikasi program utama, ia dapat menghitung kembali nilai *hash* dari program termodifikasi dan menimpa nilai *hash* lama dengan nilai *hash* baru.

Dengan sedikit perubahan, metode di atas dapat diadaptasi untuk menggunakan tanda tangan digital terlihat pada gambar 3. Langkah pertama yang harus dilakukan pembuat perangkat lunak adalah membangkitkan sepasang kunci yakni kunci publik dan kunci privat sebagai parameter untuk algoritma kriptografi kunci publik. Kunci yang dibangkitkan sebaiknya memiliki panjang tertentu untuk meningkatkan faktor keamaanan. Misalnya untuk algoritma RSA disarankan panjang kunci minimal adalah 1024-bit [7]. Dengan menggunakan kunci privat maka dapat dihitung tanda tangan digital file program.

Tanda tangan digital dan kunci publik kemudian ditambahkan di akhir file program. Walaupun demikian skema proteksi dengan menggunakan tanda tangan digital ini masih memiliki kelemahan. Kelemahan metode proteksi perangkat lunak menggunakan fungsi *hash* juga muncul pada metode tanda tangan digital. Pihak luar yang ingin memodifikasi program dapat membangkitkan kunci privat dan publiknya nya sendiri. Kemudian menghitung tandatangan digital menggunakan kunci privat miliknya (bukan milik pembuat perangkat lunak), dan mengganti kunci publik dan tanda tangan digital pada program terproteksi dengan nilai yang baru.



Gambar 3. Skema proteksi dengan menggunakan tanda tangan digital

Untuk mengatasi kelemahan tersebut, kunci publik dibuat agar tidak dapat dimodifikasi oleh pihak luar. Misalnya dengan menyimpannya, tidak dalam program, melainkan pada *server* di internet. Dengan begitu pihak luar tidak dapat menghitung tanda tangan digital yang sesuai dengan kunci publik karena ia tidak mengetahui kunci privatnya. Sehingga bila terjadi modifikasi file, akan mengakibatkan proses verifikasi gagal.

Sedangkan untuk membatasi penggunaan perangkat lunak tersebut bisa menggunakan Kode Registrasi dengan Algoritma Kunci Publik. Kode registrasi tersebut kemudian dapat dikirimkan kepada pengguna menggunakan *email* atau media pengiriman lainnya.

Berikut ini adalah contoh kode registrasi sebuah perangkat lunak yang menggunakan algoritma kriptografi RSA-2048 bit [6]:

```
rpSxwsCRTTfOsNkRBib4QjGWgSFZjvfCN1OanT
FvwqMyf43oC2+dz/wvrwUW/s7FlYg5b3+dwuJS
43cZwaXxtvQN3Sy2kcBsiyb4VNWZCX/i3PobYL
mGm0f2PIiLnTP6Uude4lMNq9BAHfqgNNEQ3sib
ob0WawrIMdlFQHsXjoOqz/TPDnTaLI7hF0em+k
jRQYD8ZbEb7uWeuOvZmndi+rI5/MzVhb9i6J7e
7Ts6NVDL1RZUgu6NwTQIzbfLTdpD3NyjYxdpCX
Ynhzy3FBD7ASKvXNSM82lu6Yl3IKPyQ/3eAQcI
SXCd3B96uff3EFyL7cPEdgP/u/IvW8kwQr1==
```

Pada contoh di atas kode registrasi terlihat panjang, sehingga sulit untuk ditulis oleh pengguna. Dengan adanya permasalaah ini, beberapa pembuat perangkat lunak memilih untuk menggunakan kunci yang pendek, sehingga dihasilkan kode registrasi yang lebih singkat. Namun tentunya langkah ini mengurangi tingkat keaamanan perangkat lunak. Berikut ini adalah contoh kode registrasi sebuah perangkat lunak yang dibuat dengan menggunakan algoritma kriptografi Elliptic-Curve 62-bit [6]:

```
JCF8T-2MG8G-Q6BBK-MQKGT-X3GBC
```

Demikian juga menurut Schertler [9], penerapan proteksi denga teknik RSA ini ada beberapa kelemahan:

1. Kunci publik dan privat adalah nilai tak bermakna (rangkaian digit yang sangat panjang)
2. Masih diperlukannya pihak ketiga yang terpercaya untuk mengkonfirmasi kunci publik
3. Masih diperlukannya infrastruktur kunci publik yang bersifat langsung / online
4. Keberlakuan kunci tidak bisa diubah

RANCANGAN SISTEM PROTEKSI *LIPIRISM@*

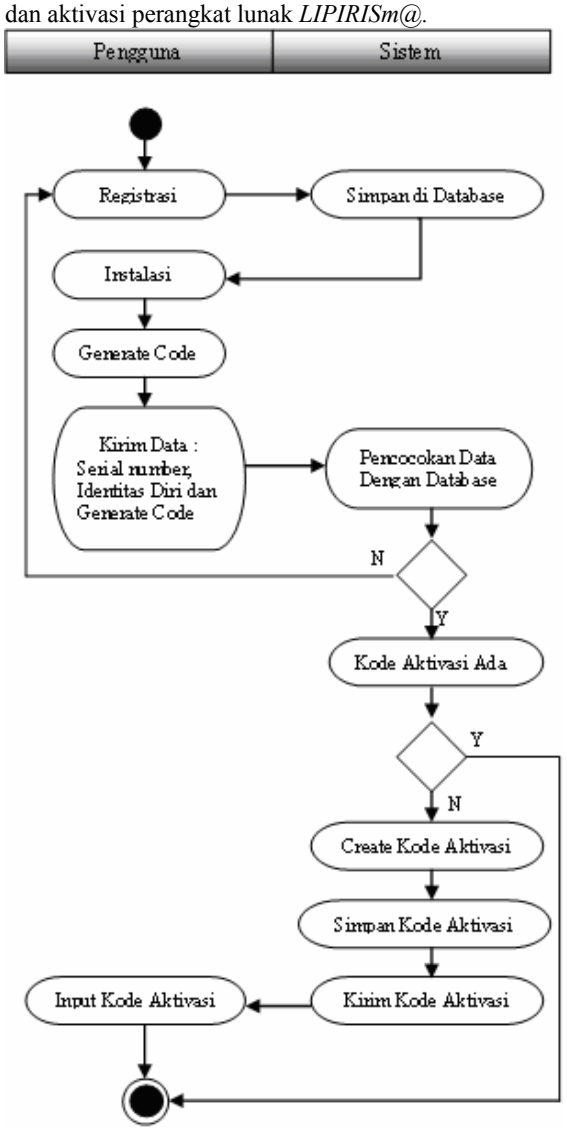
Berikut ini adalah suatu rancangan dalam mengimplementasikan proteksi terhadap perangkat lunak *LIPIRISm@*. Dimulai dengan distribusi perangkat lunak yang diikuti dengan *serial number* yang ditulis pada setiap CD yang didistribusikan.

Pada saat pengguna melakukan instalasi, maka perangkat lunak akan melakukan *generate code* sebagai tanda tangan digital, dimana kode yang digenerate ini akan selalu berbeda secara acak *(random)* setiap pengguna melakukan instalasi.

Berikutnya pengguna harus melakukan aktivasi dimana sebelumnya harus melakukan registrasi dahulu melalui internet di *http://www.lipirisma.com/registrasi.html/,* dengan mengisi formulir registrasi, dan ada beberapa kolom yang tidak boleh kosongi. Bila pengguna tidak melalukan aktivasi, maka perangkat lunak tersebut hanya bisa digunakan dengan *mode demo* dan adanya pembatasan waktu penggunaannya, agar pengguna dapat menggunakan *LIPIRISm@* secara utuh maka pengguna harus melakukan aktivasi dengan mengirimkan *serial number* dan *generate code*. Untuk sementara ini pengguna harus memasukkannya melalui situs di atas.

Sistem registrasi di www.lipirisma.com akan menerima dan melakukan pengecekan terhadap *serial number* yang dikirim dan dicocokan dengan data yang ada di database, bila cocok dan belum ada nomor aktivasinya, maka *generate code* yang dikirim akan dimasukan kedalam subprogram aktivasi dan hasilnya akan keluar nomor aktivasi dan kemudian nomor tersebut akan dikirim ke pengguna. Bagi pengguna yang ternyata *serial number* tersebut ternyata sudah ada nomor aktivasinya didalam database, maka permintaannya akan ditolak.

Berikut pada gambar 4 di bawah ini diperlihatkan *activity diagram* bagaimana rancangan proses registrasi dan aktivasi perangkat lunak *LIPIRISm@*.



Gambar 4. *Activity Diagram* proses registrasi dan aktivasi *LIPIRISm@*

KESIMPULAN

Ada beberapa pertimbangan dalam menentukan perlu tidaknya suatu perangkat lunak diproteksi, walaupun hal ini masih menjadi perdebatan dikalangan para praktisi, seperti yang diungkapkan oleh Tri Ngo [9], juga yang tercantum dalam Agenda Riset Nasional 2004-2009 dalam pengembangan teknologi informasi dan komunikasi yang yang diarahkan pada persoalan kemanan, kerahasiaan, privasi, dan integritas informasi, hak atas kekayaan intelektual, serta legalitasnya [10].
Maka *LIPIRISm@* sebagai salah satu perangkat lunak yang dibuat oleh anak bangsa ini perlu dilakukan proteksi untuk melindungi terhadap distribusinya. Selain sudah didaftarkannya sebagai salah satu perangkat lunak yang memiliki hak atas kekayaan intelektual.
Dari sekian banyak model sistem pengamanan untuk menjaga keaslian dan penggunaan perangkat lunak *LIPIRISm@* dari pengguna yang tidak berhak, maka model dengan menerapkan algoritma kriptografi kunci publik menjadi suatu pilihan, selain mudah dalam implementasinya juga tidak memerlukan biaya tambahan yang relatif besar bila dibandingkan dengan sistem dongle. Walaupun dengan sistem ini masih ada beberapa kelemahan, tetapi dibandingkan dengan sistem yang lainnya maka proteksi dengan model ini sudah mencukupi, hanya saja untuk kedepan masih perlu dicarikan suatu model lain yang bisa menutupi kelemahan tersebut.

DAFTAR PUSTAKA

[1] BSA, *Illegal PC Software use down to 84% in 2007 in Indonesia,* akses terakhir 2 September 2008 http://w3.bsa.org/indonesia/press/newsreleases/globalstudypr.cfm
[2] _____, Undang-undang no 19 Tahun 2002, http://id.wikisource.org/wiki/Undang-undang_Nomor_19_Tahun_2002, Akses terakhir 25 Agustus 2008.
[3] P.C. van Oorschot, Revisiting Software Protection?, Digital Security Group, School of Computer Science, Carleton University, Canada, Akses Terakhir 25 Agustus 2008, http://www.scs.carleton.ca/~paulv/papers/isc5.pdf
[4] Martin R Stytz, Jeff Hughes, Advancing Software Security– The Software Protection Initiative AT-SPI Technology Office AFRL/SN AFRL/SN, Akses terakhir 30 Agustus 2008 http://www.preemptive.com/documentation/SPI_software_Protection_Initative.pdf .
[5] Clark Thomborson, (2007), *Methods for Software Protection*, Keynote Address on International Forum on Computer Science and Advanced Software Technology, Jianxi Normal University
[6] Kusuma, Anugrah Redja. (2005). Proteksi Perangkat Lunak dengan Algoritma Kirptografi Kunci Publik. Bandung : ITB.
[7] Cerven, Parfol, (2002), *Crackproof Your Software*, No Scratch Press, San Fransisco
[8] Schertler M, (2005), *Identity-Based Encryption Technology Overview,* Proceeding at Internet Engineering Task Force, Baltimore
[9] Tri Ngo, Richard Sinn (2005). *The Software Protection Debate* December 19, 2005, 6.901 Final Paper Professor Robert Rines, Akses Terakhir 30 Agutsus 2008, http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-901Fall-2005/CCED0DD3-8479-48B3-800C-C39CFF9F694C/0/software_protctn.pdf
[10] Dewan Riset Nasional, (2006), *Agenda Riset Nasional 2006-2009*, Kementrian Negara Riset dan Teknologi Republik Indonesia