Pembangunan Pustaka Proteksi Perangkat Lunak dengan Algoritma RSA dan Fungsi *Hash* MD5

Denny Ranova

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : ranovaz@gmail.com

Abstrak

Perangkat lunak merupakan suatu produk yang mahal tetapi mudah untuk "dicuri" atau "dibajak". Oleh sebab itu pihak pengembang perangkat lunak harus dapat memproteksi perangkat lunak mereka. Proteksi ini meliputi pembatasan penggunaan perangkat lunak hanya oleh yang berhak dan juga penjagaan keaslian perangkat lunak. Salah satu solusi yang dapat digunakan untuk proteksi ini adalah dengan memanfaatkan tanda-tangan digital dengan fungsi *hash* dan algoritma kunci-publik. Solusi ini mencakup metode registrasi perangkat lunak untuk membatasi penggunaan perangkat lunak oleh yang berhak dan juga metode pendeteksian perubahan untuk menjaga keaslian perangkat lunak. Solusi proteksi ini dapat diimplementasikan dengan cara membuat sebuah pustaka proteksi perangkat lunak yang dapat digunakan oleh pihak pengembang perangkat lunak pada fase pembangunan *(development)* perangkat lunak. Untuk melakukan pengujian solusi ini, maka dibangun sebuah pustaka proteksi perangkat lunak yang diimplementasikan dalam bentuk kumpulan fungsi dan prosedur dengan menggunakan kakas bantu Borland Delphi 7.0. Algoritma kunci-publik yang diimplementasikan adalah algoritma RSA, sedangkan fungsi *hash* yang diimplementasikan adalah fungsi *hash* MD5. Pustaka proteksi yang dibangun mempunyai modul yang mencakup pembangkitan pasangan kunci-publik dan kunci-privat, pembangkitan kode registrasi, verifikasi kode registrasi, pembangkitan tanda-tangan digital, dan verifikasi tanda-tangan digital.

***Kata kunci**: digital signature, fungsi hash, MD5, algoritma RSA, registrasi, keaslian, kunci-publik, kunci-privat, tanda-tangan digital, pesan-ringkas.*

1. Pendahuluan

Perangkat lunak merupakan suatu produk yang mahal tetapi mudah untuk "dicuri" atau "dibajak". Oleh sebab itu perusahaan pengembang perangkat lunak harus menghadapi masalah bagaimana caranya mereka dapat menjual program yang dapat dieksekusi atau dijalankan oleh pembelinya tetapi pembeli tersebut tidak dapat mendistribusikannya lagi ke pihak lain tanpa izin dari pihak pengembang [1].

Permasalahan lain yang dihadapi oleh perusahaan pengembang perangkat lunak yaitu bagaimana menjaga keaslian perangkat lunak yang mereka kembangkan dalam proses pendistribusiannya. Menjaga keaslian perangkat lunak juga merupakan salah satu masalah yang cukup penting karena pada pendistribusiannya perangkat lunak dapat mengalami perubahan yang tidak dikehendaki baik yang tidak disengaja ataupun disengaja oleh pihak lain yang tidak bertanggung jawab.

Dari beberapa hal yang telah diungkapkan diatas sudah jelas pihak pengembang perangkat lunak harus memproteksi perangkat lunaknya agar bisa didistribusikan dengan baik. Proteksi perangkat lunak tersebut harus meliputi hal sebagai berikut :

a. Pembatasan penggunaan perangkat lunak sehingga hanya dapat digunakan oleh pihak yang berhak, misalnya pembeli perangkat lunak.
b. Menjaga keaslian perangkat lunak sehingga tidak terjadi perubahan yang tidak disengaja ataupun disengaja oleh pihak lain dalam proses pendistribusiannya.

Salah satu solusi untuk kedua hal diatas adalah dengan memanfaatkan tanda tangan digital dengan menggunakan fungsi *hash* dan algoritma kunci publik untuk proteksi perangkat lunak. Untuk mempermudah pengembang perangkat lunak memproteksi perangkat lunak yang mereka kembangkan dapat dibuat sebuah pustaka yang bisa berupa kumpulan fungsi, prosedur, objek, ataupun komponen berkaitan dengan pemanfaatan fungsi *hash* dan algoritma kunci publik untuk proteksi perangkat lunak yang dapat digunakan oleh pengembang perangkat lunak pada fase pembangunan (*development)* perangkat lunak mereka.

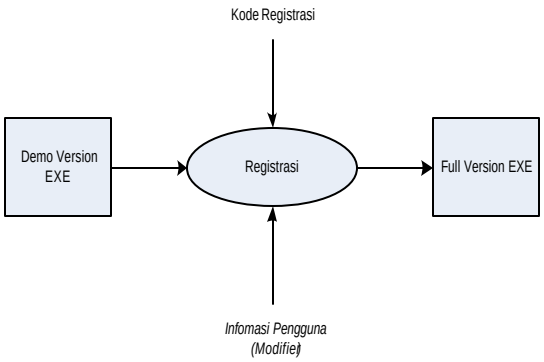
Pembangunan pustaka ini akan memberikan alternatif bagi pengembang perangkat lunak mengenai bagaimana memproteksi perangkat lunak mereka. Sebagai contoh, jika tersedia pustaka yang mendukung fitur pembatasan penggunaan perangkat lunak dengan menggunakan kode registrasi, maka pihak pengembang dapat memilih informasi apa dari pengguna yang dapat mereka gunakan untuk pembangkitan kode registrasi, apakah itu nama, email atau bahkan dapat juga menggunakan *identifier* mesin untuk membuat proteksi lebih aman. Jadi bisa dikatakan solusi berupa pustaka ini memberikan keleluasaan bagi pengembang perangkat lunak mengenai metode ataupun skema proteksi yang ingin mereka gunakan sesuai dengan kebutuhan.

2. Pemanfaatan Tanda-tangan Digital untuk Memproteksi Perangkat lunak

2.1 Registrasi Perangkat Lunak

Salah satu solusi untuk pembatasan penggunaan perangkat lunak adalah dengan menggunakan proses registrasi perangkat lunak. Dengan metode ini maka perangkat lunak hanya dapat digunakan secara penuh oleh pengguna yang telah memiliki kode registrasi. Kode registrasi ini dikeluarkan oleh pihak pengembang berdasarkan informasi yang didapat oleh pengguna.

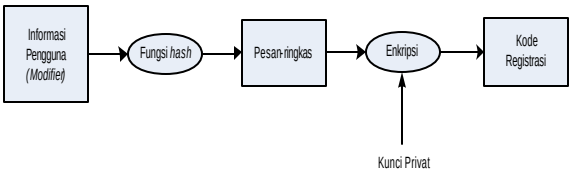
Dengan metode registrasi ini maka pihak pengembang dapat mendistribusikan perangkat lunaknya dalam bentuk versi *demo* yang dapat digunakan oleh siapapun dengan fungsi yang terbatas. Jika pengguna perangkat lunak versi *demo* ini ingin menggunakan perangkat lunak dengan fungsi secara menyeluruh maka pengguna diharuskan memiliki kode registrasi yang dapat digunakan untuk meregistrasikan perangkat lunak *demo* agar menjadi perangkat lunak yang berfungsi secara menyeluruh *(full version)*, seperti yang digambarkan pada Gambar 1.



Gambar 1 Proses Registrasi Perangkat Lunak

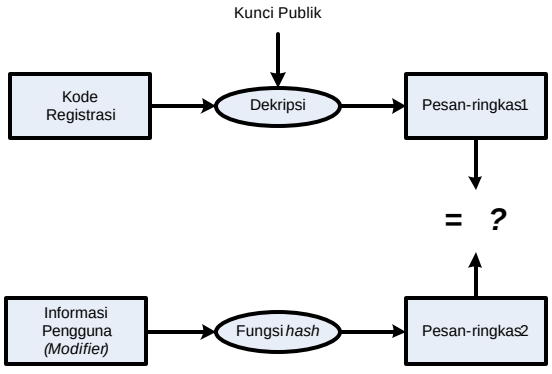
2.2 Pembangkitan dan Verifikasi Kode Registrasi

Tanda-tangan digital dapat dimanfaatkan dalam proses registrasi perangkat lunak. Hal ini bisa dilakukan dengan cara menjadikan informasi dari pengguna seperti nama, alamat, email, kode mesin komputer, ataupun nomor telepon sebagai parameter pembangkitan kode registrasi. Pembangkitan kode registrasi dilakukan dengan menganggap informasi tersebut sebagai dokumen yang akan ditandatangani dan nilai tanda-tangan digital yang dihasilkan dari informasi tersebut adalah kode registrasi, seperti yang diperlihatkan pada Gambar 2.



Gambar 2 Pembangkitan kode registrasi

Pada saat pengembangan perangkat lunak juga dibutuhkan penambahan kunci-publik pada *source* perangkat lunak yang belum dikompilasi. Kunci-publik ini digunakan untuk mendekripsi kode registrasi menjadi pesan-ringkas, sebut saja pesan-ringkas1. Lalu dilakukan pembangkitan pesan-ringkas dari informasi masukan pengguna, sebut saja pesan-ringkas2. Verifikasi kode registrasi dapat dilakukan dengan cara membandingkan pesan-ringkas1, hasil dekripsi, dengan pesan-ringkas2 yang baru saja dibangkitkan. Jika kedua nilai pesan-ringkas tersebut sama, maka artinya kode registrasi benar. Hal ini dijelaskan melalui Gambar 3.



Gambar 3 Verifikasi kode registrasi

2.2 Pendeteksian Modifikasi pada Perangkat Lunak

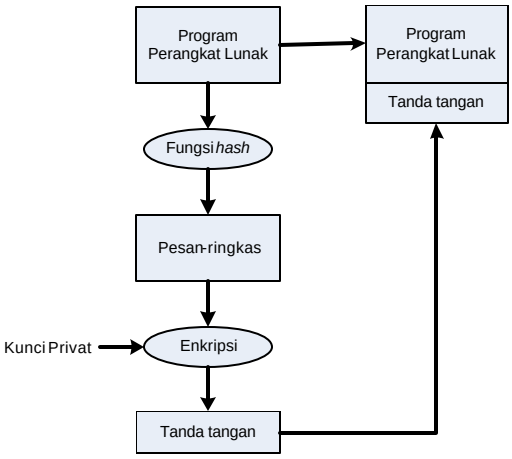
Tanda-tangan digital dapat dimanfaatkan untuk mendeteksi terjadinya modifikasi pada perangkat lunak. Pihak pengembang bisa membangkitkan

tanda-tangan dari perangkat lunaknya dan kemudian menyambungkannya pada perangkat lunak tersebut.

Langkah-langkah yang dilakukan pihak pengembang adalah sebagai berikut :

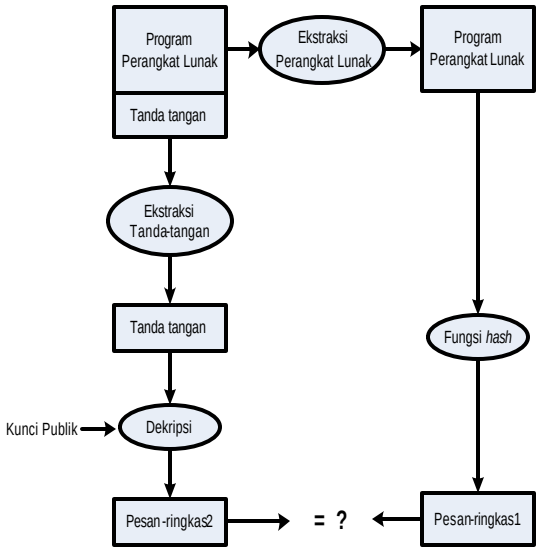
1. Membangkitkan pesan-ringkas dari perangkat lunak.
2. Mengenkripsi pesan-ringkas yang dihasilkan dengan kunci privat.
3. Menyambungkan tanda-tangan yang dihasilkan pada perangkat lunak.

Secara sederhana langkah-langkah tersebut diperlihatkan pada Gambar 4.



Gambar 4 Penyambungan tanda-tangan

Untuk proses pendeteksian modifikasi, pada saat perangkat lunak dijalankan maka perangkat lunak akan melakukan proses verifikasi tanda-tangan yang tersambung pada "dirinya". Pada proses verifikasi ini perangkat lunak akan membangkitkan pesan-ringkas dari "dirinya" sebelum tersambung dengan tanda-tangan, sebut saja pesan-ringkas1.. Lalu perangkat lunak akan melakukan dekripsi tanda-tangan yang tersambung dengan menggunakan kunci-publik yang sebelumnya sudah dimasukkan pada *source* perangkat lunak sebelum dikompilasi. Hasil dekripsi ini, sebut saja pesan-ringkas2, akan dibandingkan dengan pesan-ringkas1. Jika keduanya bernilai sama, maka artinya tidak terjadi modifikasi pada perangkat lunak tersebut, sebaliknya jika tidak sama berarti telah terjadi modifikasi. Proses verifikasi tanda-tangan ini diperlihatkan pada Gambar 5.



Gambar 5 Verifikasi tanda-tangan

3. Pustaka Proteksi Perangkat Lunak PROTLIB

PROTLIB adalah salah satu implementasi pustaka proteksi perangkat lunak yang dapat digunakan oleh pengembang perangkat lunak pada fase pengembangan perangkat lunak sehingga perangkat lunak yang dihasilkan dapat dibatasi penggunaannya dan dapat terjamin keasliannya.

Pada pustaka ini digunakan pemanfaatan tanda-tangan digital untuk memproteksi perangkat lunak, baik dalam hal pembatasan penggunaan maupun penjagaan keaslian perangkat lunak Tanda-tangan digital yang diterapkan adalah tanda-tangan digital yang menggunakan algoritma kriptografi kunci-publik dan fungsi *hash* satu-arah. Fungsi *hash* satu-arah yang digunakan adalah fungsi *hash* MD5. Sedangkan algoritma kunci-publik yang digunakan adalah algoritma RSA.

Dari kebutuhan yang ada, maka pustaka proteksi perangkat lunak PROTLIB dapat dibagi menjadi lima komponen utama, yaitu :

1. Komponen pembangkit kunci.
   Komponen pembangkit kunci digunakan oleh pihak pengembang untuk membangkitkan pasangan kunci-publik dan kunci-privat sesuai dengan algoritma kunci-publik RSA. Pasangan kunci yang dihasilkan ini akan digunakan untuk proses enkripsi dan dekripsi baik pada kode registrasi maupun tanda-tangan digital.

2. Komponen pembangkit kode registrasi.
   Komponen pembangkit kode registrasi digunakan oleh pihak pengembang untuk membangkitkan kode registrasi perangkat lunak. Pembangkitan kode registrasi ini membutuhkan kunci-privat untuk mengenkripsi pesan-ringkas dari informasi pengguna. Hasil enkripsi tersebutlah yang akan digunakan sebagai kode registrasi perangkat lunak.
3. Komponen registrasi.
   Komponen registrasi merupakan komponen yang fungsi utamanya adalah memverifikasi kode registrasi yang dimasukkan oleh pengguna. Untuk mendukung fungsi ini maka dibutuhkan fungsi pendukung seperti fungsi untuk mendekripsi kode registrasi dengan kunci-privat dan juga fungsi untuk membangkitkan file registrasi. File registrasi ini akan dibangkitkan jika registrasi berhasil.
4. Komponen pembangkit tanda-tangan digital.
   Komponen tenda-tangan digital digunakan oleh pengembang perangkat lunak untuk membangkitkan tanda-tangan digital dari perangkat lunak dengan menggunakan kunci-privat. Tanda-tangan digital ini kemudian disambungkan *(append)* pada perangkat lunak tersebut. Untuk itu, maka selain memiliki fungsi untuk membangkitkan tanda-tangan digital, komponen ini juga memiliki fungsi untuk menyambungkan tanda-tangan digital pada perangkat lunak.
5. Komponen verifikasi tanda-tangan digital.
   Komponen verifikasi tanda-tangan digital merupakan komponen yang memiliki fungsi untuk memverifikasi tanda-tangan digital yang tersambung pada perangkat lunak. Verifikasi tanda-tangan digital ini bisa digunakan untuk mendeteksi terjadinya perubahan pada perangkat lunak. Jika verifikasi tanda-tangan digital berhasil, artinya perangkat lunak masih asli dan tidak terjadi modifikasi. Sebaliknya, jika verifikasi gagal maka artinya sudah terjadi modifikasi pada perangkat lunak.

4. Kesimpulan

Kesimpulan yang dapat diambil adalah sebagai berikut :

1. Tanda-tangan digital dengan algoritma kunci-publik dan fungsi *hash* merupakan salah satu solusi yang baik untuk memproteksi perangkat lunak yang meliputi penjagaan keaslian dan pembatasan penggunaan perangkat lunak.
2. Pustaka proteksi perangkat lunak PROTLIB dapat digunakan dengan baik untuk memproteksi perangkat lunak yang meliputi penjagaan keaslian dan pembatasan penggunaan perangkat lunak.
3. Pustaka proteksi perangkat lunak PROTLIB yang diimplementasikan dalam bentuk unit-unit terkompilasi (berekstensi *.dcu) dapat digunakan atau diaplikasikan oleh pihak pengembang perangkat lunak pada fase pembangunan *(development)* perangkat lunak yang mereka kembangkan untuk menambahkan fitur proteksi.
4. Pustaka proteksi perangkat lunak PROTLIB yang memanfaatkan tanda-tangan digital, sangat peka terhadap manipulasi tanda-tangan digital, kode registrasi, informasi pengguna *(modifier)*, ataupun berkas perangkat lunak. Hal ini dikarenakan manipulasi pada hal tersebut akan mengakibatkan verifikasi tanda-tangan digital atau kode registrasi akan gagal.

Daftar Pustaka

1. Cerven, Parpol. 2002. Crackproof Your Software. No Scratch Press. SanFransisco.
2. Goldreich, O. and R. Ostrovsky. (1992). Software Protection and Simulation on Oblivious RAMs.
3. Kusuma, Anugrah Redja. (2005). Proteksi Perangkat Lunak dengan Algoritma Kirptografi Kunci Publik. Bandung : ITB.
4. Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Bandung : ITB.
5. Schneir, Bruce (1996). Applied Cryptography, Second Edition. John Wiley & Sons.