SISTEM KEAMANAN DATABASE UNTUK MEMBATASI PENGGANDAAN APLIKASI

Rangga Rahmadian Yuliendi[1]
[1]Sekolah tinggi Ilmu Komputer (STIKOM PELITA INDONESIA)
Jl.Ahmad Yani No.88 Pekanbaru Riau
Email : **rangga_rades@yahoo.com**

ABSTRAK

Penelitian ini ditujukan untuk memotivasi kepada para programer yang non opensource dalam meningkatkan kualitas dari produk atau software applikasi yang diciptakan yang anti terhadap pembajakan yang terlalu marak dilakukan oleh oknum-oknum yang tidak bertanggung jawab dalam kegiatan yang dilakukannya. Dan juga dalam mengantisipasi kerugian yang lebih besar lagi akibat kegiatan penggandaan applikasi tanpa izin.

Kata Kunci : Sistem Aplikasi Anti Penggandaan

ABSTRACT

This study aimed to motivate to the non opensource programmers in improving the quality of the product or software application that created the anti-piracy too rife done by rogue elements who are not responsible for the activities he does. And also in anticipation of even greater losses due to duplication of activities without a license application.

Keywords *: Anti Copying Application System*

1. PENDAHULUAN

Maraknya kasus pembajakan atau penggandaan yang terjadi terhadap hak cipta, khususnya terhadap produk-produk lagu, film, atau *software aplikasi* komputer memprihatinkan banyak pihak. Di Amerika hukum tentang kasus-kasus pembajakan sudah dikatakan cukup mapan untuk menjerat sang pelaku, walaupun ada beberapa pertentangan khususnya "pembajakan" untuk keperluan sendiri. www.4Shared.com adalah salah satu situs yang memberikan layanan download gratis lagu-lagu MP3 dituding telah melanggar hak cipta (*copyright*). Walaupun pihak www.4Shared.com berdalih apa yang mereka lakukan, dengan memberikan layanan download gratis lagu-lagu, adalah sah karena mereka tidak menarik bayaran sedikitpun atau tidak melakukan perdagangan. Hal demikian dilindungi oleh undang-undang, demikian pihak www.4Shared.com memberikan sanggahan.

Contoh di atas adalah salah satu contoh kasus yang banyak mendapat sorotan, dan menjadi perhatian dunia sebagai sebuah pelajaran bagaimana menegakkan hukum perindungan hak cipta. Salah satu teknik yang digunakan dalam memproteksi *software* selain dari undang-undang hukum adalah teknik proteksi kata kunci (*Password*). Namun teknik ini tidak terlalu memproteksi pada sebuah *software* dalam realitanya teknik pemberian *password* pada sebuah *aplikasi software* yang menggunakan database bisa saja dijebol atau dibocorkan oleh orang yang mengetahui *password* tersebut kepada orang lain. Contoh kasusnya seperti Toko A membeli *Software Aplikasi* Sistem Informasi ke seorang Programer dengan nominal Rp.500.000, selanjutnya Toko A juga menjual *Software Aplikasi* Inventori tersebut ke Toko B tanpa sepengetahuan dari programer dengan harga relatif rendah dari sebelumnya, dan kata kunci (*password*) dari admin oleh Toko A juga dibocorkan ke pada Toko B.

Yang menarik sebenarnya, ini bisa diatasi dengan negosiasi gaya Indonesia. Yaitu dengan menggunakan pendekatan secara langsung terhadap pembajak *software* Indonesia untuk mau mendukung *software* lokal, dengan menawarkan sharing penjualan yg menarik, tapi dengan syarat tidak menjual bajakannya. Sudah ada beberapa vendor *software* lokal melakukan hal tersebut, dan sepertinya cukup efisien.

UU HAKI 19 Tahun 2002 sudah diundangkan, tapi pelaksanaan masih belum menggembirakan. Dilema juga terjadi karena secara ekonomi, masyarakat akan kesulitan membeli *software-software propietary*. Kampanye penggunaan *software* legal perlu tetap diteruskan, masyarakat sebaiknya diberikan pilihan dan solusi dalam kampanye penggunaaan *software* legal tersebut:

1. Beli lisensi *software* apabila ingin menggunakan *software* propietary
2. Selain itu silakan gunakan *software* opensource maupun freeware

2. DASAR TEORI

Tantangan yang paling besar dihadapi oleh industri computer yang berkaitan dengan hukum adalah pembajakan *software*. Pembajakan *software* adalah memperbanyak dan mendistribusikan hasil *copy* suatu *software* secara illegal. Masalah pembajakan ini merupakan isu ekonomi, di mana pendapatan dan sebuah hak kepemilikan kekayaan intelektual seseorang hilang atau dicuri. Pembajakan menghadirkan situasi yang kompleks dan rumit yang merupakan dilemma besar dalam era informasi (*information age*). Karena alat yang kita gunakan (komputer) sebagai sumber informasi merupakan alat yang sama yang digunakan orang lain untuk mencuri informasi tersebut, dalam hal ini yaitu *software*.

Mungkin kita bisa menangkap pembajak - pembajak profesional, tapi bagaimana dengan jutaan orang-orang yang membajak dan menggunakan hasil bajakan tersebut. Untuk menanggapi hal ini, beberapa perusahaan telah menerapkan *software* atau *hardware* beradasarkan skema proteksi (*protection schemes*). Contoh dari penerapan hal ini adalah mengikat sebuah *dongle* dengan sebuah program. *Dongle* adalah komponen *hardware* yang kecil yang dimasukkan ke dalam PC, USB, dan sumber - sumber pemasukkan data lainnya. Dan ada juga dengan cara teknik pemberian *password*.

Isu legal juga merupakan masalah yang semakin diper buruk dengan isi *licensing agreement*, apalagi bagi tempat - tempat yang menggunakan banyak komputer seperti sekolah dan kantor. Karena berdasarkan *licensing agreement*, *software* hanya dapat digunakan oleh satu orang. Untuk mengatasi hal ini, Borland internasional, sebuah perusahaan *software* besar tidak menggunakan *licensing agreement* untuk produk - produknya. Mereka memperlakukan sebuah program sama seperti buku, jadi banyak orang dapat menggunakannya, namun *software* tersebut hanya milik satu orang dan dalam PC yang sama.

Bagian dari masalah ini bersifat filosofis. Ketika sistem hukum melindungi *software* sebagai bagian dari *intellectual property*, tetapi elemen filosofisnya kurang bekerja sempurna. Kita harus mengenali dan menerima dasar filosofis dibalik ide - ide kepemilikan sebelum kita mengikuti prosedur atau garis hukum yang berlaku. Sebaliknya jika ukuran-ukuran pencegahan dilegalkan namun tidak diperkuat secara penuh, maka situasinya akan berlanjut.

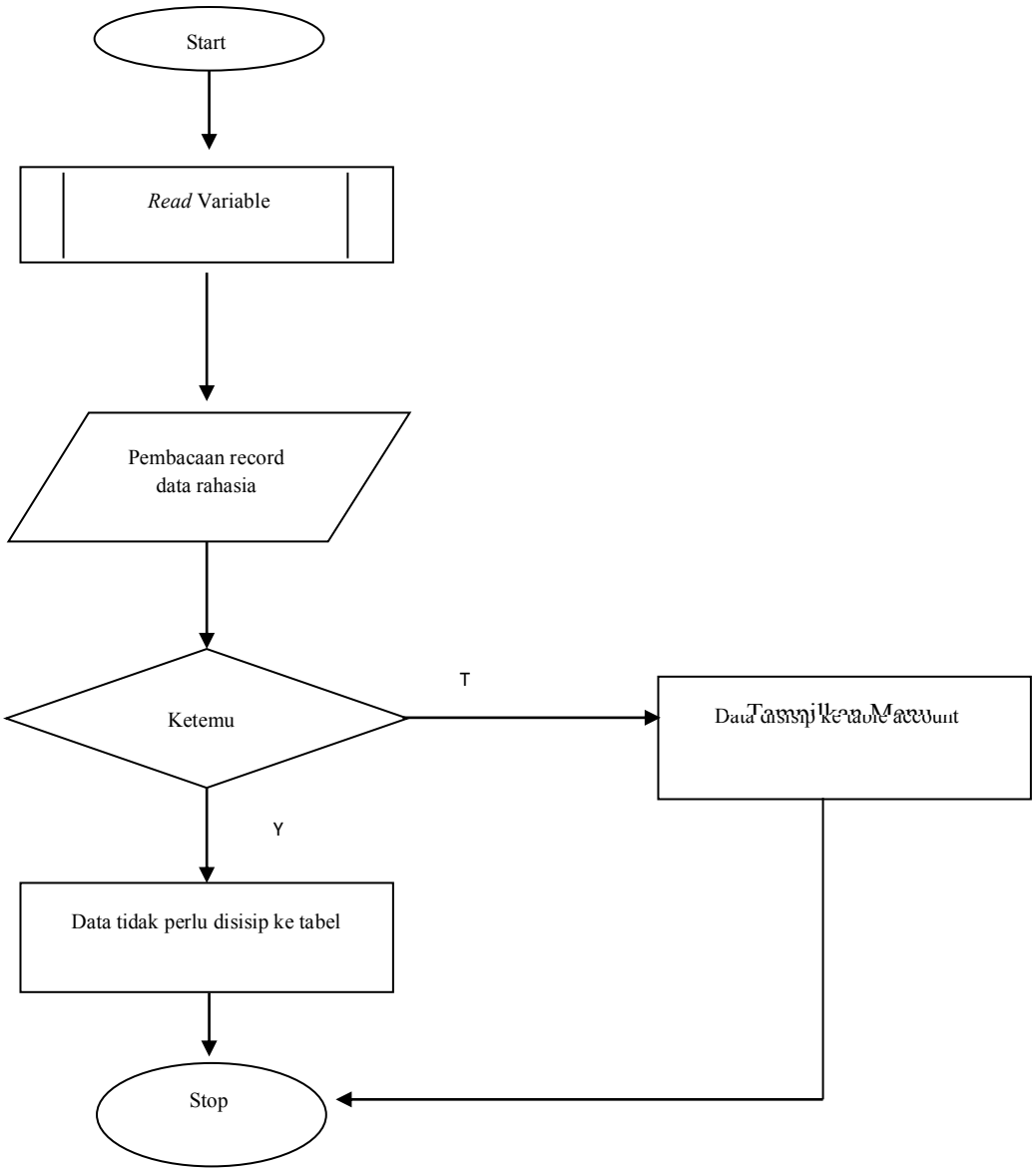
Dalam bidang komunikasi, selain *software*, film, CD, bahkan pelayanan TV kabel pun dibajak.

3. PEMBAHASAN

Pada bagian ini menguraikan tentang langkah-langkah perancangan dan pembuatan sistem *aplikasi* sistem informasi anti penggandaan atau pembajakan dengan teknik pembacaan terhadap *record* data rahasia pada sebuah tabel di dalam database engine, Yang mana teknik yang akan digunakan adalah *steganorecord* dengan metoda *embbeding* atau bisa juga disebut *injection* yaitu teknik menyuntikan langsung sebuah data rahasia ke dalam sebuah arsip atau media. Dan setelah data rahasia disisipkan program *aplikasi* yang dirancang akan diberi perintah (*Syntax*) untuk melakukan pembacaan terhadap data rahasia tersebut sebagai *authentifikasi* dari aplikasi itu sendiri untuk tujuan perizinan dalam pemakaian *aplikasi* yang tanpa batasan.

3.1 Diagram Alir

Diagram alir yang digunakan dalam penyisipan *record* data rahasia (*Steganorecord*) secara otomatis akan digambarkan dalam bentuk flow diagram seperti terlihat pada gambar 3.1.



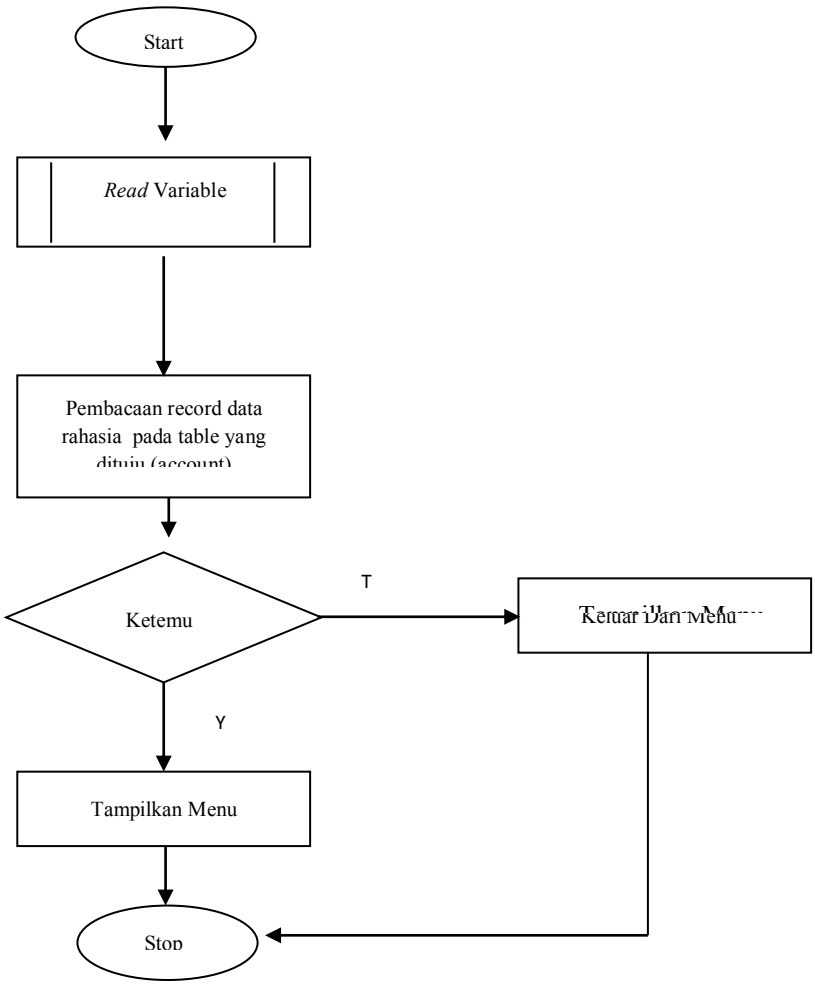
Gambar 3.1. Diagram Alir Penyisipan *Record* Data Rahasia

Pada gambar diagram alir di atas jalannya proses penyisipan *record* data rahasia dilakukan pembacaan terhadap *record* data yang sama terlebih dahulu.

1. Apabila tabel yang ditujukan telah berisi data maka proses penyisipan tidak perlu dilakukan ke *aplikasi* tersebut karena aplikasi telah ada data atau teregister lebih jelasnya.
2. Dan sebaliknya apabila tabel yang ditujukan dalam penginputan *record* data rahasia masih kosong maka penyisipan data rahasia tersebut secara otomatis akan terisi ke dalam tabel yang ditujukan sebagai media tempat pembacaan *record* data

rahasia sebagai *authentifikasi* terhadap sistem *aplikasi* tersebut apabila dijalankan.

Diagram alir yang digunakan dalam pembacaan *record* data rahasia (*authentifikasi*) akan digambarkan dalam bentuk flow diagram seperti terlihat pada gambar 3.2.



Gambar 3.2. Diagram Alir Pembacaan *Record* Data Rahasia

Secara garis besar proses yang dilakukan sebuah sistem yang telah diproteksi dapat diurutkan sebagai berikut :

1. Perintah pertama melakukan pembacaan terhadap *record* data rahasia yang ada pada *database engine* yang telah tersedia dari *software*. Pembacaan ini dilakukan pada

saat program dijalankan di mana *syntax* dari program tersebut diletakkan pada tampilan awal program atau menu program.

2. Jika pada pembacaan terhadap database bernilai kosong atau tidak ada data sama sekali maka menu dari program tersebut akan mengeluarkan pesan bahwa program tidak bisa dijalankan alias program belum teregister dan tampilan menu program akan hilang dari layar monitor komputer.

3. Apabila pembacaan terhadap data rahasia pada database yang dituju memiliki *record* data maka tampilan menu program akan bisa dijalankan dan program bisa digunakan sebagaimana mestinya.

Database yang digunakan dalam pembacaan ini adalah database bawaan atau *Database Engine* dari hasil penginstalan dari *software* program yang digunakan. Biasanya jumlah data pada database ini bernilai kosong yaitu tidak ada data. Dengan memanfaatkan kondisi seperti ini pada database kita bisa merancang sebuah *aplikasi* yang tidak akan berjalan dengan semestinya alias terkunci, hal ini dikarenakan database yang dibaca masih memiliki jumlah *record* nol (0) dan dengan menyisipkan sebuah data rahasia dengan teknik *Steganorecord* maka *aplikasi* dapat dijalankan setelah melewati proses pembacaan terhadap data rahasia terlebih dahulu.

3.2. Penyajian Fakta dan Aturan

Berikut adalah *rule-rule* yang digunakan berdasarkan kriteria yang ada dalam pembacaan *record* data rahasia (*authentifikasi*). Dalam hal perizinan pemakaian dari sistem *aplikasi* atau atau dibolehkan atau tidaknya sebuah sistem ini dipakai cukup menggunakan 2 (dua) daftar aturan (*Rule*) yaitu kondisi yang menyatakan perizinan, ter *authentifikasi* atau *teregister* dan aturan yang menyatakan tidak diizinkan, non *authentifikasi* atau *non register*.

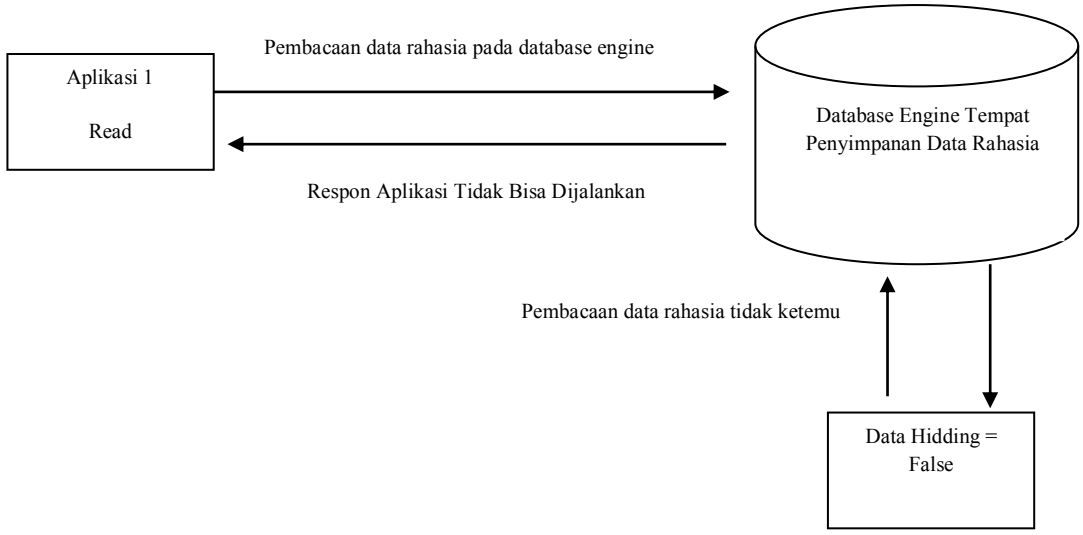
Tabel 3.1 Daftar Aturan (*Rule*)

No	Aturan (*Rule*)
1.	JIKA Data Rahasia = True AND Nilai <> 1 MAKA Program tidak bisa dijalankan (Tidak Teregister)
2.	JIKA Data Rahasia = True And Nilai = 1 MAKA Program bisa dijalankan (Teregister)
3.	JIKA Data Rahasia = False And Nilai <> 1 MAKA Program tidak bisa dijalankan (Tidak Teregister)
4.	JIKA Data Rahasia = False And Nilai = 1 MAKA Program tidak bisa dijalankan (Tidak Teregister)

3.3 Teknik Yang Digunakan

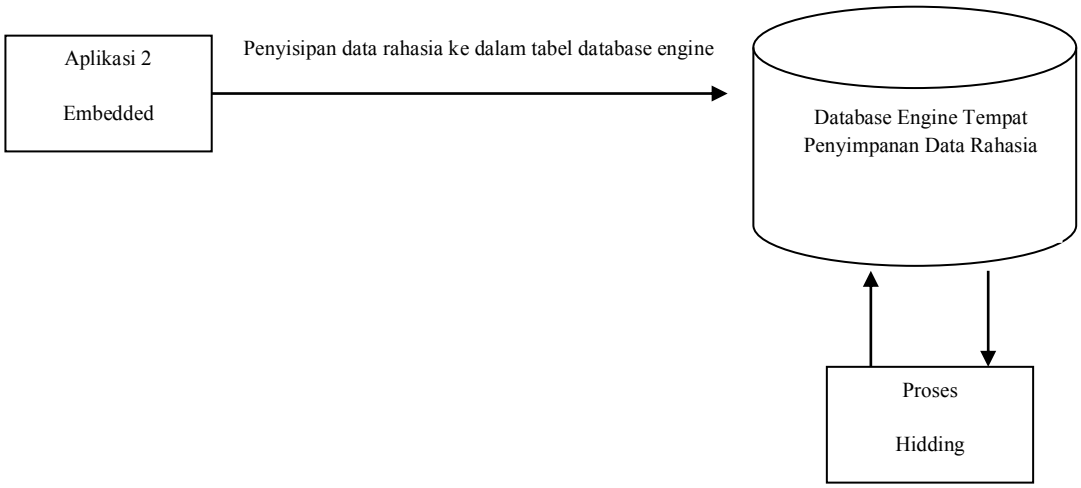
Teknik yang digunakan dalam membatasi penggandaan atau pembajakan dari *aplikasi* sistem informasi di sini adalah teknik Steganografi di mana konsep kerja dari teknik ini adalah menyisipkan sebuah nilai atau data rahasia yang disipkan secara

otomatis ataupun secara manual ke dalam database engine. Yang mana nantinya data *record* rahasia tersebut akan dibaca oleh tampilan awal *aplikasi* sistem sebagai proses untuk meminta perizinan dalam pemakaian *aplikasi* itu sendiri pada CPU. Apabila database engine masih bernilai *Default* maka secara otomatis *aplikasi* tidak bisa dijalankan. Untuk menjalankan *aplikasi* dengan sempurna tanpa batasan, *database engine* terlebih dahulu disisipkan sebuah data rahasia yang mana data rahasia tersebut berfungsi untuk meautentifikasi *aplikasi* saat *aplikasi* tersebut dijalankan. Dan perintah pembacaan data rahasia tersebut dilakukan di saat program aplikasi dijalankan.
Perhatikan Gambar 3.3., Gambar 3.4., Gambar 3.5. :



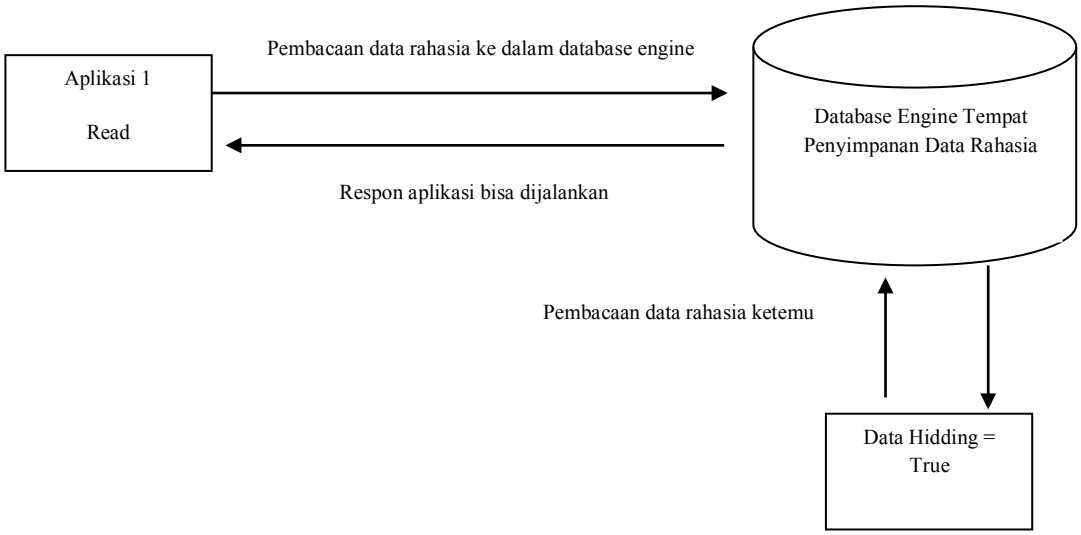
Gambar 3.3. *Aplikasi* 1 Melakukan Pembacaan Terhadap Data Rahasia Pada Database Engine

Keterangan Pada Gambar 3.3. di sini terlihat bahwa *aplikasi* pertama saat dijalankan akan melakukan pembacaan terhadap data rahasia di dalam *database engine*, pada saat kondisi awal *database engine* masih bernilai nol (0). ini dimanfaatkan oleh keamanan dari *aplikasi* dengan memberikan respon terhadap user bahwa penggunaan *aplikasi* ditolak. Hal ini dikarenakan bahwa *aplikasi* di saat dijalankan secara diam-diam melakukan pembacaan terhadap data yang disisipkan ke dalam *database engine*, apabila data rahasia tersebut tidak ditemukan di dalam database maka respon dari *aplikasi* itu sendiri akan melakukan penolakan.



Gambar 3.4. Proses Penyisipan Data Rahasia (*Embedded*) ke Database Engine

Keterangan dari Gambar 3.4. pada gambar di atas terlihat bahwa sebuah aplikasi sedang melakukan penyisipan data rahasia (*Embedded*) ke dalam sebuah *database engine*, tujuan dari *aplikasi* kedua tersebut adalah untuk memenuhi permintaan awal *aplikasi* pertama sebagai salah satu syarat saat pembacaan terhadap data rahasia di saat *aplikasi* pertama dijalankan. Sehingga *aplikasi* yang pertama berhasil mendeteksi adanya sebuah data rahasia di dalam *database engine* dan *aplikasi* pertama tersebut berhak untuk digunakan oleh pemakai.



Gambar 3.5. *Aplikasi* 1 Berhasil Membaca Data Rahasia Pada Database Engine

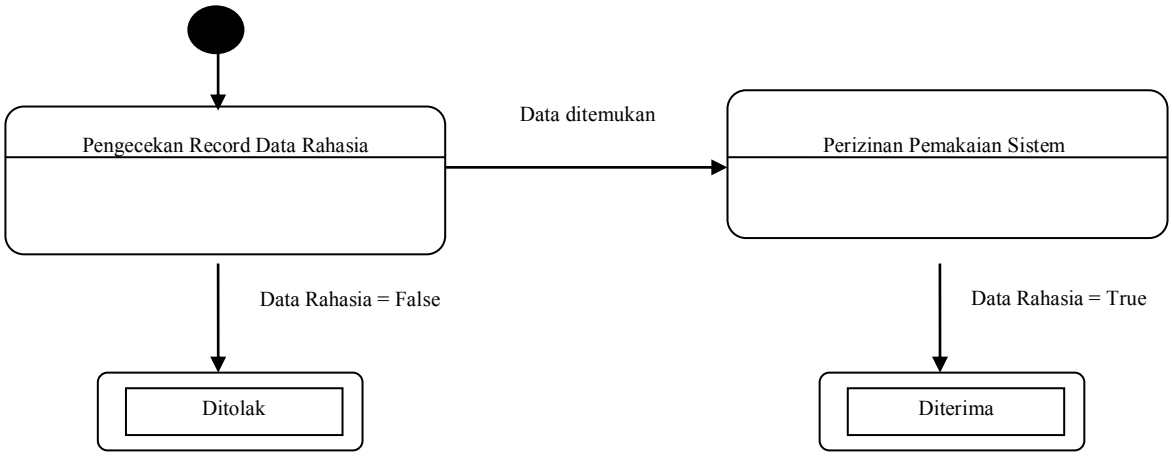
Keterangan dari gambar 3.5. dari gambar di atas dapat dilihat bahwa *aplikasi* pertama berhasil menemukan data rahasia yang disisipkan (*Embedded*) secara *steganorecord* sewaktu melakukan pembacaan terhadap data rahasia dalam sebuah *database engine* sehingga *aplikasi* tersebut mendapat respon untuk dapat digunakan sepenuhnya.

4. RANCANGAN

Pada rancangan sistem ini menggunakan diagram proses State Diagram Status Diagram digunakan untuk menguraikan perilaku suatu sistem. Diagram Status menguraikan semua tahapan yang mungkin dari suatu objek ketika peristiwa terjadi. Masing-Masing diagram yang pada umumnya menghadirkan objek dari kelas tunggal dan menjejaki tahapan yang berbeda tentang objeknya melalui sistem. Penggunaan diagram status untuk mempertunjukkan perilaku dari suatu objek melalui banyak kasus dari penggunaan sistem itu. Hanya menggunakan diagram status untuk kelas di mana diperlukan untuk memahami perilaku objek sampai keseluruhan sistem. Tidak semua kelas akan memerlukan suatu diagram status dan diagram status tidaklah bermanfaat untuk gambarkan kerja sama atau kolaborasi dari semua objek di dalam suatu kasus penggunaan.

Diagram di bawah menunjukkan suatu super-*state*. Kedua-Duanya Pemeriksaan dan Pengiriman tahapan dapat di transisi ke dalam Status yang dibatalkan, maka suatu transisi ditunjukkan dari suatu super-*state* nama aktif Kepada status pembatalan. Sebagai pembanding, status Pengiriman hanya dapat melakukan transisi kepada Status yang dikirimkan, maka kita menunjukkan suatu panah hanya dari pengirim status kepada Status yang dikirim.

Gambar 4.1. menunjukkan *state*chart diagram pembacaan *record* data rahasia oleh sistem yang digunakan dalam permodelan penulisan *Jurnal* ini.



Gambar 4.1 Super *State* Diagram

5. PENGUJIAN SISTEM

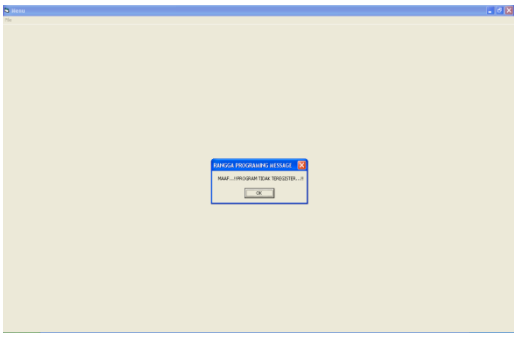
Pengujian Terhadap Komputer 1

Setelah menu dari applikasi sistem informasi dirancang, pengujian terhadap menu dari applikasi sistem informasi harus dilakukan terlebih dahulu, apabila syntax atau perintah pembacaan data pada salah satu tabel pada database yang dituju bernilai nol (0) maka tampilan menu dari applikasi sistem informasi akan hilang.

Pengujian Terhadap Komputer 2

Apabila pada computer sebelumnya (komputer 1) applikasi sistem informasi yang kita rancang bisa berjalan dengan sempurna dengan bantuan Operasi analisis, disini penulis menyebutnya dengan sistem Crack. Sekarang dilanjutkan dengan pengujian terhadap komputer yang berbeda (komputer 2). pada komputer 2 ini applikasi dari sistem informasi tadi di copy kan dengan menggunakan flasdisk dari komputer 1 dan dipaste kan pada komputer dua pada posisi dan nama folder yang persis sama. Pada komputer 2 jumlah record pada tabel di dalam database awal dari software DML (*Data Manipulating Language*) masih bernilai nol (0). Nilai ini bernilai konstan atau tetap apabila software DML (*Data Manipulating Language*) baru diinstalkan pada komputer 2 tersebut. Selanjutnya dilakukan pengujian terhadap menu pada komputer 2 sebelum sistem Crack dijalankan. Apabila disaat menu dijalankan terjadi penolakan, hal ini disebabkan jumlah record pada tabel didalam database awal terhadap komputer 2 masih bernilai nol (0) dan perancangan sebuah applikasi sistem informasi anti penggandaan atau pembajakan telah berhasil diciptakan sesuai dengan perencanaan.

Aplikasi Yang Belum Teregister atau aplikasi yang tidak menemukan pembacaan terhadap data rahasia yang disisipkan akan menampilkan pesan tersaji pada gambar berikut ini.



Gambar 5.1 Proses Pengujian komputer

5. PENUTUP

Berdasarkan penelitian dan pembahasan yang telah dilakukan sebelumnya, maka kesimpulan yang dapat diambil adalah sebagai berikut :

Aplikasi Anti Penggandaan dengan menggunakan metode steganography dan akan membantu tingkat security terhadap sebuah sistem aplikasi yang bersifat komersil (*Copyright* dan berbayar).

DAFTAR PUSTAKA

Ariyus, Dony, 2006. *Computer Security*. Yogyakarta: Penerbit ANDI.

Ariyus, Dony, 2005. Kamus Hacker . Yogyakarta: Penerbit ANDI.

Maya, *Steganografi LSB* (http://maya9luthu.blogsome.com/2006/12/11/steganografi-lsb/ , diakses 6 April 2007)

Sukrisno, *Steganografi*,(http://mysukris.blogspot.com/2007/ 04/steganografi-eureka-i-found-it.html, diakses 6 April 2007)

Pindo, *Steganografi dengan Media Plaintext* (http://orangsakti.blogsome.com/2006 /12/14/steganography-dengan-media-plaintext/, diakses 6 April 2006)

Pindo, *Steganografi pada Plains Text* (http://orangsakti.blogsome.com/2007/01/05/steganografi-pada-plains-text/, diakses 6 April 2007)

Wikipedia, Kriptografi (http://id.wikipedia.org/wiki/Kriptografi, 6 April 2007)