Majalah Ilmiah Matematika & Komputer, April 2007 Nomor 1/Tahun XXIII
ISSN 0216-4728

APLIKASI AGEN BERGERAK UNTUK PENDETEKSIAN PENYUSUP PADA JARINGAN KOMPUTER

Adi Kristanto
Didin Mukodim
Universitas Gunadarma

ABSTRAK

Perkembangan jaringan komputer memerlukan pengelolaan yang baik oleh administrator jaringan. Permasalahan yang sering terjadi dalam jaringan komputer adalah masalah keamanan. Salah satunya adalah masalah penyusupan dalam jaringan komputer. Untuk menangani adanya penyusupan diperlukan suatu Sistem Pendeteksi Penyusup (SPP) SPP melakukan deteksi penyusupan dalam jaringan komputer.

Agen bergerak merupakan salah satu klasifikasi dari agen perangkat lunak yang memiliki kemampuan untuk bergerak dari suatu tempat ke tempat lain, dan secara mandiri melakukan tugas di tempat barunya tersebut, dalam lingkungan jaringan komputer.

Tulisan ini membahas tentang teknologi agen bergerak untuk melakukan deteksi penyusupan dalam Local Area Network (LAN). Agen bergerak akan melakukan proses pengumpulan data kondisi jaringan yang terdistribusi. Dari data tersebut kemudian akan disimpan di basis data dan dapat dianalisis oleh administrator jaringan dalam mengambil keputusan berdasarkan informasi yang diterima.

Aplikasi ini memanfaatkan Aglet Perangkat lunak Development Kit (ASDK) untuk membangun aplikasi agen bergerak dan bahasa pemrograman Java. Penggunaan agen bergerak sebagai Sistem Deteksi Penyusupan dapat dengan cepat dalam memperolah data yang ada di setiap komputer.

***Kata kunci**: Agen bergerak, Sistem Deteksi Penyusupan, Jaringan Komputer, Java.*

PENDAHULUAN

Perkembangan jaringan komputer telah membawa peningkatan yang sangat berarti bagi lalu lintas komunikasi data baik dari segi skalabilitas, reliabilitas maupun teknologi yang diterapkan.

Tugas pengelolaan jaringan yang dilakukan administrator jaringan memiliki banyak permasalahan, di antaranya yang berkaitan dengan keamanan jaringan komputer. Keamanan jaringan komputer yang baik dapat menyediakan jaminan mengenai ketersediaan jaringan, kerahasiaan, integritas, dan ketersediaan suatu objek seperti data, proses, atau jasa. Lemahnya keamanan jaringan komputer dapat menimbulkan adanya penyusupan.

Penyusupan adalah usaha seseorang untuk merusak atau menyalahgunakan sistem, atau setiap usaha yang melakukan kompromi integritas, kepercayaan atau ketersediaan suatu sumber daya komputer. Untuk menangani adanya penyusupan perlu adanya suatu Sistem Pendeteksi Penyusup (SPP). SPP adalah sistem komputer (bisa merupakan kombinasi perangkat lunak dan perangkat keras) yang berusaha melakukan deteksi penyusupan. SPP akan mendeteksi adanya penyusupan atau sesuatu yang dianggap sebagai mencurigakan atau tindakan ilegal dan melaksanakan tindakan ketika suatu penyusupan dideteksi. SPP tidak melakukan pencegahan ketika terjadinya penyusupan. Pengamatan untuk melakukan pemberitahuan itu bergantung pada bagaimana melakukan konfigurasi SPP.

Suatu jaringan komputer harus selalu dijaga keamananya oleh administrator jaringan agar bebas dari penyusupan. Di samping itu, seorang administrator jaringan tidak mungkin sepanjang waktu selalu berada di dekat jaringan komputer yang dikelolanya. Hal ini memberikan kesulitan dalam menangani jaringan komputer yang tidak

dapat ditangani oleh administrator jaringan. Untuk itu diperlukan suatu *Agen* yang dapat melakukan tugas-tugas pengelolaan keamanan jaringan komputer, dan dapat beroperasi melalui jaringan internet atau intranet. Agen perangkat lunak adalah entitas perangkat lunak yang didedikasikan untuk tujuan tertentu. Agen bisa memiliki ide sendiri mengenai bagaimana menyelesaikan suatu pekerjaan tertentu. Salah satu agen perangkat lunak yang menangani masalah jaringan adalah agen bergerak. Agen bergerak adalah agen yang aktif dan dapat bergerak menuju ke komputer lain, atau menjelajahi jaringan untuk menjalankan tugasnya.

LANDASAN TEORI

Sistem Pendeteksi Penyusup (SPP)

SPP merupakan perangkat lunak atau perangkat keras yang melakukan automatisasi proses monitoring kejadian yang muncul di sistem komputer atau jaringan, menganalisanya untuk menemukan permasalahan keamanan. SPP adalah pemberi sinyal pertama jika seorang penyusup mencoba membobol sistem keamanan komputer kita. Secara umum penyusupan bisa berarti serangan atau ancaman terhadap keamanan dan integritas data, serta tindakan atau percobaan untuk melewati sebuah sistem keamanan yang dilakukan oleh seseorang dari internet maupun dari dalam sistem. SPP tidak dibuat untuk menggantikan fungsi *firewall* karena kegunaannya berbeda.

Sebuah sistem *firewall* tidak bisa mengetahui apakah sebuah serangan sedang terjadi atau tidak, sedangkan SPP mengetahuinya. Dengan meningkatnya jumlah serangan pada jaringan, SPP merupakan sesuatu yang diperlukan pada infrastruktur keamanan di kebanyakan organisasi. Secara singkat, fungsi SPP adalah pemberi peringatan kepada administrator atas serangan yang terjadi pada sistem.

Agen perangkat lunak

Agen perangkat lunak (selanjutnya di sebut agen saja) adalah entitas perangkat lunak yang didedikasikan untuk tujuan tertentu. Agen bisa memiliki ide sendiri mengenai bagaimana menyelesaikan suatu pekerjaan tertentu atau agenda tersendiri.

Agen perangkat lunak memiliki fungsi, peran, dan perbedaan mendasar bila dikaitkan dengan perangkat lunak program yang ada. Atribute dan karakteristik yang dimiliki oleh agen perangkat lunak saat ini adalah otonomi, intelijen, alasan, dan pembelajaran, bergerak dan statis, delegasi, reaktivitas, proaktif dan berorientasi tujuan, serta mempunyai kemampuan berkomunikasi dan berkoordinasi. Pada dasarnya daftar karakteristik dan atribut yang ada merupakan hasil survei dari karakteristik yang dimiliki oleh agen yang ada pada saat ini.

Agen bergerak

Agen bergerak adalah agen yang aktif dan dapat bergerak menuju komputer lain, atau mejelajahi jaringan untuk menjalankan tugasnya. Agen bergerak sering digunakan untuk mengumpulkan data, informasi atau suatu perubahan. Agen bergerak tidak terikat pada sistem dimana ia mulai dieksekusi. Agen bergerak mempunyai kemampuan unik untuk memindahkan dirinya sendiri dari satu sistem ke sistem yang lain dalam suatu jaringan. Kemampuan untuk berkeliling memungkinkan agen bergerak untuk berpindah ke sistem yang mengandung objek yang akan berinteraksi dengan agen dan kemudian mengambil manfaat selama berada dalam *host* dan jaringan yang sama dengan objek.

Secara praktis dapat dikatakan bahwa agen bergerak adalah sebuah program yang dapat menghentikan eksekusi, berjalan melalui jaringan dengan membawa kode dan status-nya, dan kemudian melanjutkan eksekusinya pada *host* yang lain. *Aglets* merupakan contoh perangkat lunak yang memungkinkan pengembangan dan penerapan agen bergerak ini. Beberapa penerapan dari agen bergerak adalah pengumpulan data, pencarian dan

penyaringan, pemantauan *asinkron*, dan pemrosesan paralel.

Aglets

Aglets adalah objek java yang dapat bergerak dari satu *host* ke *host* lain dalam suatu jaringan. *Aglet* yang sedang bekerja disuatu *host* dapat menghentikan eksekusinya, pergi ke *host* yang lain dan kemudian memulai eksekusinya kembali. Ketika *aglet* bergerak, aglet membawa kode program dan juga *status* dari semua objek yang membawanya. Sebuah mekanisme keamanan yang terpasang akan mengamankan *host* dari *aglet* yang tidak dapat dipercaya.

Istilah *aglet* sesungguhnya adalah kombinasi dari kata *agen* dan *applet*. Perbedaan dengan *applet* adalah *aglet* juga membawa serta status, dan memiliki *itinerary* (rencana perjalanan). ASDK (*Aglets Perangkat lunak Development Kit*) adalah paket perangkat lunak yang digunakan untuk menulis aplikasi *agen bergerak*. ASDK menggunakan bahasa java dan bisa didapatkan secara gratis dari internet. Karena kemampuannya membuat *lightweight agen*, maka *aglet* API ini sering disebut sebagai RISC agen bergerak.

Tahiti adalah suatu program aplikasi yang bekerja sebagai *server aglet*. Aplikasi *tahiti* merupakan suatu paket dengan ASDK. Beberapa *server* (tahiti) dapat dijalankan dalam satu komputer dengan cara memberikan nomor *pangkalan* yang berbeda. *Tahiti* menyediakan *pengguna interface* untuk pemantauan, penciptaan, pengiriman, dan pemusnahan suatu aglet serta untuk menetapkan hak akses untuk server agen.

METODE PENELITIAN

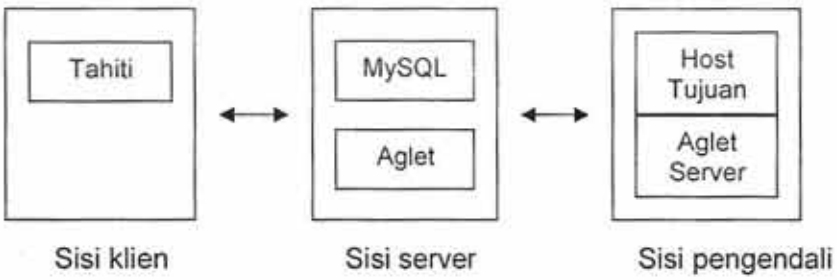
Perangkat yang dibutuhkan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak. Perangkat lunaknya adalah Aglets API Development Kit, yang digunakan untuk membuat agen perangkat lunak, Java Development Kit sebagai bahasa pemrograman dan MySQL sebagai basis data server. Pengimplementasian dilakukan dalam lingkungan sistem operasi *Microsoft Windows*.

Selain itu, penelitian juga dilakukan dengan mengadakan uji coba program untuk membuktikan kemampuan aplikasi yang telah dibuat, terutama cara kerja aplikasi agen bergerak. Uji coba dilakukan di *Local Area Network* (LAN) laboratorium dengan topologi bintang.

HASIL DAN PEMBAHASAN

Perancangan Sistem Deteksi Penyusupan

Sistem deteksi penyusupan terdiri dari 3 blok diagram, yaitu sisi klien, sisi server, dan sisi pengendali. Sisi klien berupa *Tahiti* sebagai antarmuka dengan pengguna. Sisi server terdiri dari aglet server dan basis data server di *host* asal. Sedangkan sisi pengendali berupa *aglet* server pada *host* tujuan yang menjadi tujuan deteksi penyusupan serta informasi kondisi *host* tersebut yang akan diambil. Gambar 1 menunjukkan diagram blok arsitektur sistem.



Gambar 1 Diagram blok arsitektur sistem

Agen bergerak ini nantinya beroperasi pada sumber data yang terdistribusi. Jika seorang administrator ingin melihat kondisi di setiap *host*, maka agen bergerak yang akan dikirimkan ke setiap *host* yang akan melakukannya. Di sini sistem deteksi penyusupan dirancang sebagai sensor eksternal, dengan pengumpulan data dari sejumlah *host* (berbasis *multi host*), dan sumber data diambil baik secara langsung maupun tidak.

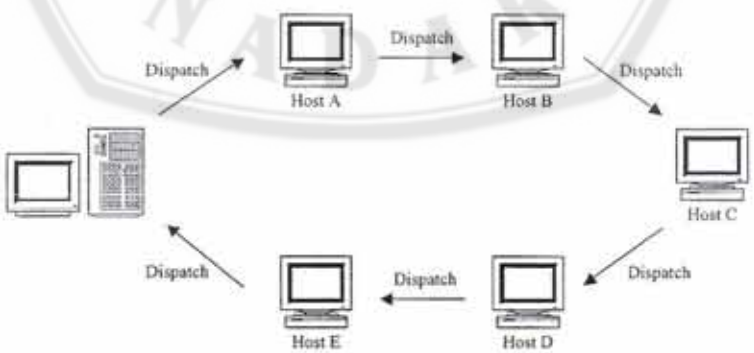
Agen bergerak disini merupakan HIDS (Host sistem pendeteksi penyusup) yang terdistribusi, sehingga monitoring dilakukan secara tidak langsung. Hal ini dilakukan karena:

1. Kebanyakan penyusupan bisa dideteksi pada *host*, misal eksekusi perintah, akses ke pelayanan, dan lain-lain. Serangan akan berakhir pada *host* meski melalui jaringan, misalkan saja *flooding* jaringan juga akan terdeteksi di *host*.
2. Memungkinkan pengumpulan data yang merefleksikan secara akurat apa yang terjadi, ketimbang menebak paket yang lewat jaringan.
3. Dalam lalu lintas jaringan yang tinggi, paket bisa terlewat oleh monitor jaringan (NIDS).
4. Bisa menentukan tingkatan dan jenis aktivitas tertentu secara spesifik yang ingin dimonitor.

Sedangkan monitoring secara langsung dilakukan karena sumber data tidak langsung yang telah diubah oleh penyusup, tidak semua kejadian terekam, sumber data tidak langsung menyebabkan volume data yang besar, sehingga pemrosesannya membutuhkan lebih banyak waktu dan sumber daya, monitoring langsung hanya mengambil data yang diperlukan saja, dan monitoring tidak langsung mengalami permasalahan skalabilitas.

Agen bergerak akan mengambil sejumlah informasi dari masing-masing *host* dalam jaringan komputer, di antaranya nama *host*, nama pengguna, direktori tempat penyimpanan berkas, pangkalan yang aktif, arsitektur sistem operasi, nama sistem operasi, versi sistem operasi, ruang memori, dan waktu pengambilan informasi.

Urutan langkah deteksi penyusupan suatu jaringan komputer terdiri dari beberapa proses. Di bawah ini akan dijelaskan urutan langkah proses deteksi penyusupan dari suatu jaringan komputer tersebut. Dengan asumsi setiap *host* sudah dijalankan *aglet server* yang siap menerima *aglet* yang masuk pada pangkalan yang sudah ditentukan. Sementara *tahiti* dijalankan pada komputer asal dimana administrator akan melihat hasil kerja agen tersebut. Gambar 2 menunjukkan skema proses deteksi penyusupan pada jaringan komputer.

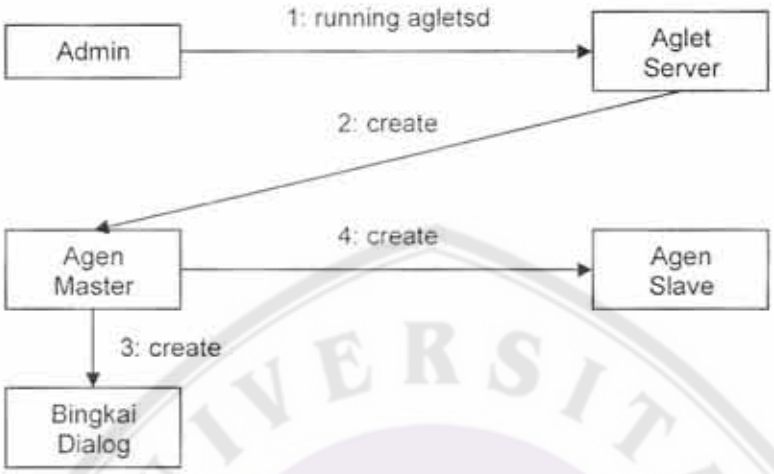


Gambar 2 Skema proses deteksi penyusupan

Perancangan Agen

Dalam aplikasi ini dipergunakan dua buah agen yaitu agen master (Finger) dan *agen slave* (FingerSlave). Agen master sebagai agen statis, akan dijalankan di komputer asal oleh administrator

mempergunakan *Tahiti*. Setelah agen master berhasil dipanggil, maka *agen* tersebut akan membuat *bingkai* untuk dialog interaksi dengan *pengguna*, serta membuat *agen SlaveIDS*. Gambar 3 menunjukkan kolaborasi diagram penciptaan *agen*.



Gambar 3 Kolaborasi diagram penciptaan agen

Pembuatan Kelas

Aplikasi agen bergerak ini terdiri dari tiga buah kelas utama, yang bertugas untuk melakukan deteksi dan interaksi masukan/keluaran dengan pengguna. Kelas tersebut adalah Finger, FingerWindow,

Kelas Finger merupakan kelas utama yang akan dijalankan pertama kali oleh pengguna dari antarmuka *Tahiti*. Kelas *Finger* merupakan agen statis yang mengontrol masukan dari kelas *FingerWindow*, serta bertugas membentuk, mengirimkan/menerima informasi dari kelas *FingerSlave*.

Kelas FingerWindow merupakan kelas yang bertujuan membuat tampilan dialog untuk interaksi dengan pengguna. Sejumlah pekerjaan yang dilakukannya:

1. Menerima masukan penambahan dan pengurangan *host* yang akan menjadi tujuan perjalanan agen bergerak (*itinerary*), serta perintah pengiriman agen.
2. Menampilkan informasi deteksi yang diperoleh oleh agen bergerak, dan menyimpan informasi deteksi ke dalam basis data.
3. Menampilkan status keberangkatan/ kepulangan suatu agen bergerak.

Hal pertama yang dilakukan sebelum agen bergerak berjalan adalah memanggil berkas *itinerary* (rencana perjalanan agen). Berkas ini berisi nama daftar *host* yang akan dikunjungi oleh agen melalui kelas *FingerSlave*. Pengguna dapat mengisi berkas tersebut sebelum agen dijalankan.

FingerSlave adalah kelas agen bergerak yang akan berkeliling jaringan. Kelas ini dibuat oleh kelas *Finger*, dikirimkan ke *host* tujuan, mengambil informasi dari *host*, dan melanjutkan perjalanan sesuai rencana perjalanan (*itinerary*). Sejumlah pekerjaan yang dilakukan oleh *kelas SlaveIDS*:

1. Melakukan inisialisasi yang diperlukan untuk memulai perjalanan, seperti rencana perjalanan. Program akan melakukan pembacaan *berkas itinerary* sebelum *agen* dijalankan.
2. Melakukan perjalanan ke *host-host* tujuan. *Agen* akan berjalan di dalam jaringan komputer dari satu *host* ke *host* yang lain yang berada di dalam rencana perjalanan (*itinerary*).
3. Mengambil informasi pada *host* tersebut. Setiap *host* yang dikunjungi,

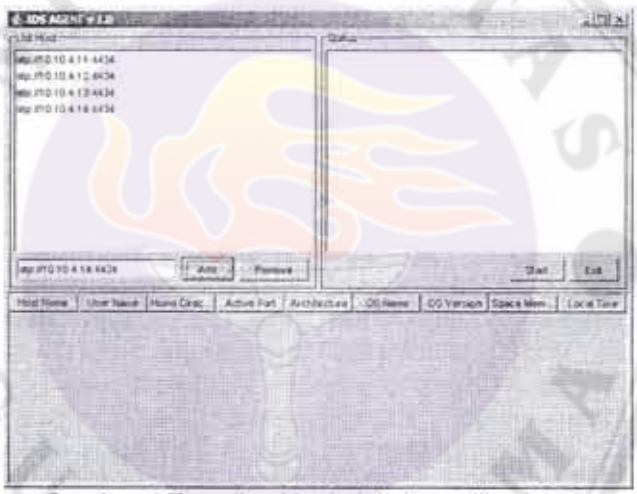
Agen akan mengambil informasi yang dibutuhkan. Jika terdapat *host* yang tidak aktif atau tidak terhubung ke dalam jaringan komputer, *agen* akan menampilkan pesan ke dalam tampilan aplikasi SPP.

4. Kembali ke *host* asal pengiriman, memberikan pesan kepulangan ke *kelas Finger*. Setelah *host* berhasil mengambil informasi dari setiap *host*, *agen* akan kembali ke kelas *Finger* atau aplikasi SPP.
5. Menampilkan informasi hasil perjalanan dan penyimpanan hasil ke *basis data*. Hasil dari perjalanan agen akan ditampilkan di aplikasi SPP oleh kelas *FingerWindow*. Hasil yang ditampilkan akan langsung tersimpan pula ke dalam *basis data*.

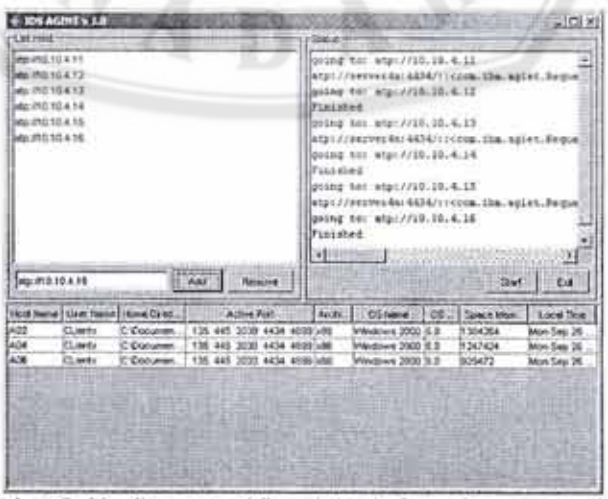
PEMBAHASAN

Perancangan Aplikasi

Tampilan dari aplikasi ini terdiri atas 3 bingkai, yaitu bingkai daftar *host* sebagai tempat untuk memasukkan alamat *host*, bingkai status sebagai status perjalanan agen dan bingkai tampilan untuk menampilkan hasil dari perjalanan agen ke masing-masing *host*. Gambar 4 menunjukkan tampilan program yang digunakan sebagai masukkan *host* dan status dari agen bergerak. Sedangkan Gambar 5 menunjukkan hasil perjalanan agen ke masing-masing host.



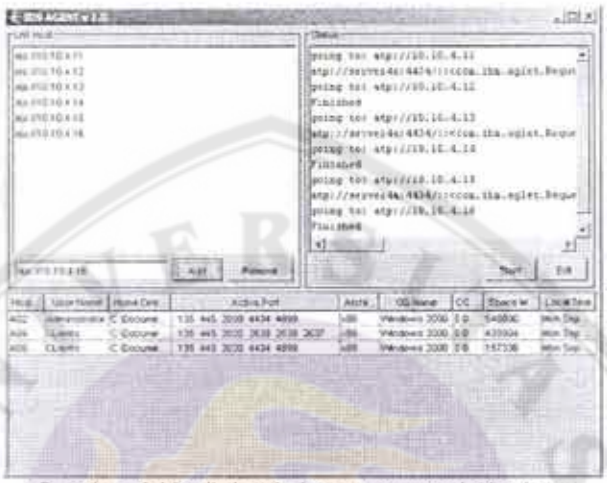
Gambar 4 Tampilan bingkai dialog aplikasi SPP.



Gambar 5 Hasil pengambilan data dari masing-masing host.

Hasil Uji Coba

Pengujian deteksi penyusupan dilakukan dengan melakukan penyalahgunaan *privilege* atau hak akses dari pengguna klien menjadi administrator. Hasil informasi yang diperoleh dapat diketahui, bahwa ada *host* yang memiliki pengguna sebagai sesorang administrator. Gambar 6 menunjukkan hasil deteksi penyusupan yang dilakukan oleh klien dengan meningkatkan hak aksesnya menjadi administrator.



Gambar 6 Hasil deteksi penyusupan hak akses

PENUTUP

Kesimpulan

Penggunaan aplikasi agen bergerak dalam pembuatan SPP memberikan beberapa kelebihan, di antaranya beban jaringan yang sangat rendah karena pemrosesan data dilakukan secara lokal di *host* jaringan. Agen bergerak mampu menangani skalabilitas jaringan. Bila terdapat elemen baru dalam jaringan, agen bergerak mampu menggandakan dirinya ke mesin baru di jaringan.

Pengujian Sistem Deteksi Penyusupan terbukti dapat mengenali perilaku anomali pengguna ketika sedang menggunakan fasilitas jaringan komputer. Aplikasi *SPP* agen mempunyai beberapa kelebihan, yaitu penggunaan sumber daya memori yang cukup kecil, sehingga tidak mengganggu proses lainnya dalam sistem operasi. Proses pengambilan data yang cepat dan terkoneksi dengan *basis data*, sehingga administrator jaringan dapat menentukan kebijakan jaringan dengan cepat.

Aplikasi SPP tidak dapat mencegah terjadinya penyusupan pada *host* secara langsung karena pada dasarnya sistem deteksi penyusupan hanya melakukan proses *monitoring* kejadian yang muncul di sistem komputer atau jaringan, menganalisisnya untuk menemukan permasalahan keamanan.

Saran

Dalam perkembangannya, sistem deteksi penyusupan telah dirancang dengan bermacam-macam algoritma dan struktur deteksi. Diawali dengan berbasis *host*, lalu berubah ke sistem berbasis jaringan, dan dalam beberapa tahun kemudian memiliki kecenderungan kombinasi terdistribusi dari keduanya. Bagaimanapun, selama perubahan itu, sumber informasi yang dipergunakan tidak berubah. Diharapkan hasil kerja ini akan memberikan suatu panduan untuk integrasi pada perancangan sistem selanjutnya yang akan berdampak pada peningkatan kinerja.

Kemungkinan pengembangan teknologi agen bergerak berikutnya adalah:

1. Menyatakan properti dari agen bergerak berkaitan dengan penggunaannya untuk deteksi penyusupan, dan arsitektur untuk

pembuatan SPP berdasar pada agen bergerak.

2. Merancang suatu prototip dari SPP mempergunakan arsitektur tersebut, sehingga menunjukkan kemungkinan mempergunakan agen bergerak untuk melakukan deteksi penyusupan yang dapat dijalankan di *platform* sistem operasi berbeda.
3. Agen bergerak bisa meningkatkan kemampuan deteksi sistem, mengeksplorasi kemampuan deteksi serangan tertentu yang susah bila diimplementasikan dengan SPP tradisional. Dengan sekumpulan *agen* ini menembus batasan tradisional dari SPP dengan berbasis host atau berbasis jaringan.

DAFTAR PUSTAKA

Admir Kulin, *A Distributed Security Managament System Based on Agen bergeraks*, Technischen Universitat Wien, Wina, 2001.

Danny B Lange, *Agen bergeraks with Java: The Aglets API*, www.moe-lange.com/danny/wwwj.pdf, California, 2000.

_____, *Programming and Deploying Java Agen bergeraks with aglets*, Addison-Wesley, New York, 1998.

_____, *Mobile Object and Agen bergerak: The Future of Distributed Computing*, General Magic Inc, California, www.moe-lange.com/danny/ecoop98.pdf, California, 2000.

InfoLinux, *Sistem Pendeteksian Intrusi*, www.infolinux.web.id, edisi juni 2002, Jakarta, 2002

Joel Scambray, Stuart and G. kurtz. *Hacking Exposed: Network Security Secrets and Solutions*, McGraw-Hill, New York, 2001.

Mitsuru Oshima, Guenter, *Aglets Specification 1.1 Draft*, IBM Corp, www.trl.ibm.co.jp/aglets/spec11.html, Tokyo, 2000.

Naughton, Patrick., *The Java Handbook, Konsep Dasar Pemrograman Java*, Osborne/McGraw-Hill Book Co. dan Penerbit ANDI Yogyakarta, Yogyakarta, 2001.

Qusay H. Mahmoud, *Distributed Programming with Java*, Manning Publications Co, Greenwich, 2000.

Rebecca Bace and Petter Mell, *Sistem pendeteksi penyusup*, NIST Special Publication on SPP, California, 2002.