

PARADIGMA MOBILE AGENT DALAM MANAJEMEN JARINGAN KOMPUTER

Adang Suhendra, Ruddy J. Suhatrik

Universitas Gundarma, Jakarta

E-mail: adang@staff.gunadarma.ac.id, ruddyjs@staff.gunadarma.ac.id

Abstrak

Sistem manajemen jaringan komputer dapat meyakinkan kita bahwa pengoperasian system jaringan beroperasi secara normal dengan mengawasi aktifitas pengguna jaringan, status sumber daya jaringan dan konfigurasi system. Teknologi mobile agent memberikan suatu kerangka kerja yang dapat mengatasi kekurangan pada teknologi manajemen jaringan komputer terpusat yang bersifat rendah fleksibilitas dan konfigurasi dimana proses pengumpulan dan analisis data jaringan melalui transfer data yang sangat besar sehingga menyebabkan fasilitas bandwidth menjadi penuh sehingga memperlambat throughput. Mobile agent dapat diimplementasikan menggunakan salah satu teknologi dasar: mobile code atau remote objects. Kedua metode tersebut menerapkan konsep distribusi proses yang diatur dari kendali terpusat. Kerangka kerja mobile agent menyediakan fasilitas yang mendukung semua model agent, yaitu model komputasi, model keamanan, model komunikasi dan juga termasuk model navigasi. Model agent termasuk layanan siklus hidup agent seperti membuat, menghapus, memulai, menunda dan menghentikan agent. Mobile agent pada pemanfaatannya dalam manajemen jaringan digunakan untuk: mengumpulkan informasi dari beberapa nilai MIB sebagai indikator kesehatan jaringan, mengambil informasi manajemen jaringan dari table SNMP, dan penyaringan konten SNMP berdasarkan ekspresi filter yang beragam. Paradigma mobile agent dalam melakukan manajemen jaringan telah menjadi salah satu solusi teknologi yang dapat memberikan hasil manajemen yang komprehensif dan bersifat sangat fleksibel tanpa membenani bandwidth jaringan dalam transmisi data hasil inspeksi serta konsep distribusinya dapat menangani proses manajemen untuk system jaringan dengan ukuran yang terus berkembang dengan cepat.

Kata kunci: mobile agents, network management, SNMP.

1. Pendahuluan

Pada dasarnya manajemen sistem jaringan komputer adalah melakukan pemantauan dan kendali ke peralatan yang terhubung ke jaringan dengan mengumpulkan dan menganalisa data yang berhubungan dengan sistem jaringan dari peralatan tersebut. Dengan berkembangnya model sistem jaringan komputer yang sudah demikian heterogen, manajemen jaringan menjadi suatu hal yang semakin sulit. Bahkan adanya peningkatan permintaan pengguna akan kualitas dan kehandalan sistem jaringan merupakan hal yang semakin menjadi lebih sulit lagi, karena membutuhkan pengaturan sistem jaringan yang lebih handal.

Pada kebanyakan manajemen sistem jaringan yang konvensional masih berdasarkan pada teknologi SNMP (Simple Network Manajemen Protocol) yang memberikan fleksibilitas bagi administrator jaringan dalam mengatur network secara keseluruhan dari satu lokasi. Tetapi seperti diketahui SNMP merupakan sistem manajemen jaringan komputer yang terpusat yang memiliki beberapa kekurangan yaitu: terjadinya *bottleneck* arus data di pusat pengawasan, kurang dalam mengantisipasi skalabilitas sistem jaringan, berdampak pada *load* pemrosesan yang berlebihan di pusat pengaturan, penggunaan *bandwidth* jaringan yang berlebihan untuk aksi manajemen jaringan.

Salah satu alternatif pendekatan yang dapat mengatasi kekurangan model SNMP adalah menggunakan model pengaturan secara

terdistribusi, yaitu melalui sistem manajemen yang interoperability. Model manajemen terdistribusi menjawab permasalahan pada model manajemen terpusat, tetapi model ini juga memiliki kekurangan seperti keterbatasan skalabilitas dan mekanisme koordinasi yang kompleks antar tempat pengaturan. Solusi teknologi mobile agent merupakan mekanisme gabungan antara sistem terpusat dan terdistribusi.

Mobile agent adalah suatu objek software khusus yang otonomus dan memiliki kemampuan untuk migrasi dari satu node ke node lainnya, membawa data dan logika, melakukan suatu aksi dari user. Sistem manajemen jaringan berbasis mobile agent menggunakan agent dan kemampuan manajemen jaringan yang dapat mengatur peralatan (*node*) setelah migrasi ke node tersebut. Mobile agent dapat diimplementasikan dengan menggunakan dua teknologi: mobile code atau object jarak jauh.

Paradigma mobile agent telah diimplementasikan dalam suatu kerangka kerja *mobile agent* atau *mobile agent framework (MAF)* [1]. Beberapa aplikasi *mobile agent* dalam lingkup fungsional OSI dari manajemen jaringan [2] mengilustrasikan bagaimana *mobile agent* dapat membantu untuk menyelesaikan permasalahan suatu sistem manajemen yang berdasarkan pada komponen-komponen stasioner. Teknologi yang digunakan dari sistem *mobile agent* adalah *Java Virtual Machine (JVM)*, dan yang terbaru adalah software Jini [7].

2. Mobile Agent

2.1 Pengertian Dasar Agent

Suatu *agent* adalah pergerakan, perpindahan ataupun aktifitas objek bergerak yang dikirim ke system jaringan oleh kerja proses suatu *client* yang dilakukan atas ijin pengguna yang telah ditentukan sebelumnya. Biasanya *agent* mengunjungi beberapa layanan spesifik untuk melakukan beberapa tugas atau perintah. Dalam kaitannya dengan partisipasi dalam system *mobile agent*, suatu penyedia layanan menjalankan satu atau lebih *agent server* pada beberapa node komputer yang ada di suatu jaringan. Proses ini bertanggung jawab sebagai *hosting* dan mengeksekusi setiap *agent* yang tiba di jaringan. Komunikasi antar *Agent server* menggunakan protokol *server-to-server*. Fungsi utama dari protokol ini adalah melakukan transfer suatu *agent* secara aman dari satu server ke server lainnya.

Sebagaimana pengertian *agent*, *Mobile agent* adalah suatu *software agent* yang dapat bergerak atau berpindah antar lokasi-lokasi. Sebagai tambahan terhadap model dasar *agent*, setiap *software agent* mendefinisikan *model siklus hidup*, *model komputasi*, *model keamanan*, dan *model komunikasi*. [3]. Suatu *mobile agent* juga dikarakteristikan sebagai *model navigasi*. *Mobile agent* dapat diimplementasikan dengan menggunakan satu dari dua teknologi dasar: *mobile code* [4, 5] atau *remote objects* [6].

Agar *mobile agent* dapat digunakan, komponen pada suatu sistem jaringan harus bekerjasama dan menyatu ke dalam suatu kerangka kerja mobilitas (*mobility framework*). Kerangka kerja tersebut harus menyediakan fasilitas yang mendukung semua *model agent*, termasuk di dalamnya model navigasi. Sedangkan untuk model siklus hidup merupakan layanan untuk membuat, menghapus, memulai, menunda, menghentikan suatu *agent*. Model komputasi yang dilakukan suatu *agent* adalah kemampuan menghitung termasuk manipulasi data dan pengendalian ancaman/gangguan. Model keamanan menggambarkan cara bagaimana *agent* dapat mengakses sumber daya jaringan, termasuk juga cara mengakses internal suatu *agent* dari jaringan. Model komunikasi mendefinisikan komunikasi antara *agent* dengan *agent* dan antara *agent* dengan entitas lain (seperti peralatan pada suatu jaringan). Pada transportasi suatu *agent* (dengan atau tanpa *state*-nya) diantara entitas, dua komputasi yang berada pada lokasi yang berbeda dikendalikan oleh model navigasi. Sehingga dari masing-masing model tersebut secara jelas, akan memerlukan biaya seperti untuk peningkatan kebutuhan memori dan penundaan (*delay*) eksekusi serta akses pada setiap peralatan di sistem jaringan.

Ukuran *mobile agent* tergantung pada apa yang akan dikerjakan, seperti dalam *swarm*

intelligence, ukuran *agent* sangatlah kecil, namun konfigurasi atau pendiagnosaan *agent* menjadi sangat besar, karena harus melakukan pengkodean algoritma yang kompleks atau *reasoning engines*. Namun bagaimanapun, *agent* tersebut dapat memperbesar kemampuan *on-the-fly*, *on-site* dengan men-*download* kode-kode yang diperlukan dalam jaringan. Beberapa penerapan *mobile agent* adalah pengumpulan data, pencarian dan penyaringan, pemantauan *asynchron*, dan pemrosesan paralel.

Keuntungan dari penerapan *mobile agent* adalah:

- Mengurangi beban jaringan melalui pengemasan suatu aplikasi dan mengirimkannya ke *host* tujuan melalui interaksi secara local juga pengurangan. Selain itu *mobile agent* juga dapat mengurangi aliran data pada jaringan. Data yang berjumlah besar akan disimpan di *remote host*, dan data tersebut akan diproses di area local, daripada mengirimkannya melalui jaringan.
- Efisiensi sumber daya. Konsumsi sumber daya (CPU dan memori) dapat dihemat, sebab *mobile agent* menetap dan bekerja hanya pada satu *node* pada satu waktu. *Node* yang lain tidak menjalankan *agent* sampai *node* tersebut memerlukannya.
- Menanggulangi jaringan *latency*. Sistem *real-time* yang kritis perlu tanggap terhadap perubahan lingkungannya secara *real-time*. Keterlambatan tanggapan yang diakibatkan oleh masalah jaringan harus dihindari. Masalah tersebut diatasi oleh *mobile agent* dengan mengirim *agent* ke tujuan dan dieksekusi secara local.
- *Encapsulate Protocol*. *Mobile Agent* mereduksi ketidak praktisan *up-grading* protokol pada pengiriman data di system terdistribusi dengan cara mengirim *agent* ke *remote host* dengan tujuan membentuk *channel*, yang berdasarkan protokol yang sudah dibakukan sebelumnya.
- Eksekusi paralel. Komputasi masif dibagi ke dalam sejumlah *agent*, dilakukan *dispatch* ke node yang paling sesuai untuk eksekusi dari tiap komponen, dan merakitnya di *home*. Sumber daya *hardware* bukan lagi jadi batasan.

Keuntungan *mobile agent* lainnya, antara lain: Skalabilitas, Paradigma Komputasi yang Adaptif, Beradaptasi secara Dinamis, Mendukung lingkungan yang Heterogen, *Real-Time Notification* dan Andal dan Toleran terhadap Kesalahan.

2.2 Kerangka kerja Mobile Agent: Standarisasi

Seperti kebanyakan aktivitas komunikasi, standarisasi merupakan hal yang paling penting, tak terkecuali *mobile agent*. *Open Management Group (OMG)* telah membuat rancangan standarisasi

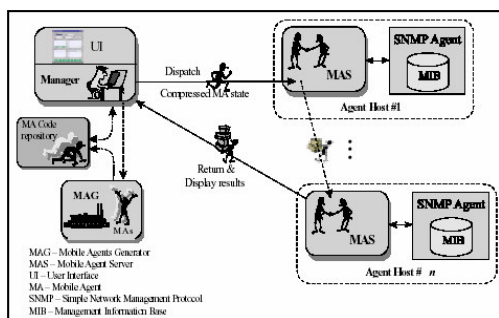
untuk *mobile agent*. Rancangan tersebut mempunyai konsentrasi pada platform yang netral, dan setiap *chunk* bahasa penidentifikasi kode *mobile* dan lingkungan eksekusi.

Proposal yang ditawarkan *OMG* mengidentifikasi kebutuhan area *mobile code*, dengan *gateway* diantaranya, yang menyediakan virtual layer suatu aplikasi *agent*.

Fasilitas *mobile agent* mencakup storage dan *backup* suatu *agent*, transfer pembuatan *remote agent* dan penempatan metode *agent*. Rancangan tersebut juga menekankan pada *CORBA*, dengan *IOP* sebagai protokol transportasi, dan menunjukkan bahwa banyak layanan pradefinisi *CORBA* akan digunakan untuk mendukung aktivitas *mobile agent* (misal, penamaan).

2. Mobile Agent Untuk Manajemen Jaringan

Sistem manajemen jaringan menghadapi hal yang terpeting dalam *software agent* seperti proliferasi/perkembangan data dan lingkungan yang heterogen. Suatu kesalahan harus didiagnosa dengan cepat dan diperbaiki secara otomatis atau operator manusia perlu diberitahu dan dinformasikan dengan arahan yang tepat untuk melakukan suatu aksi/kegiatan. Dalam jaringan yang besar, operator harus berinteraksi secara *remote* dengan banyak peralatan dari pusat pengendali. Untuk mengakomodasi keberagaman komponen-komponen jaringan membutuhkan suatu aplikasi manajemen yang terhubung dan banyak antar-muka dan peralatan. Bentuk arsitektur kerangka kerja *mobile agent* diilustrasikan pada gambar 1. berikut ini:



Gambar 1. Kerangka Kerja Mobile Agent

Fungsi dari model manajemen OSI

Ada lima kategori fungsional dari model manajemen OSI yang diperlukan untuk mengelola komunikasi jaringan, yaitu: manajemen kesalahan, manajemen accounting, manajemen konfigurasi, manajemen kinerja dan manajemen keamanan.

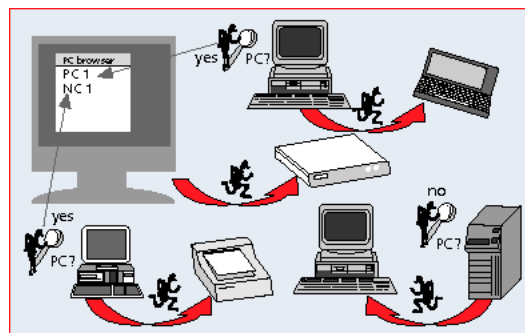
Dalam manajemen jaringan, pencarian otomatis merupakan salah satu fungsi dasar sistem manajemen. *Mobile code* merupakan suatu alat yang dapat melakukan perintah-perintah pencarian [8]. *Mobile code* tidak dapat digunakan untuk pencarian jaringan yang dasar (*node discovery*

alone). Teknik pencarian yang sering digunakan adalah mengirim pesan *ping* ke alamat IP pada suatu domain.

Selain itu, suatu *mobile agent* yang disebut sebagai *deglet* (setelah suatu pendelegasian *agent* [9]) dapat dibuat dengan satu perintah mengidentifikasi *node* yang dikunjungi. Kemudian *deglet* tersebut dimasukkan ke dalam jaringan dan berjalan untuk mengimplementasikan pola-pola migrasi [10]. Ada beberapa cara yang serupa seperti algoritma *ping*, namun metodenya lebih fleksibel, dimana *interlocator* tidak memerlukan pengetahuan mengenai jaringan.

Mengakhiri suatu perintah dapat ditentukan secara heuristic dalam *deglet*, sebagai contoh, dengan menghitung *hop* atau mencari nilai rata-rata jumlah kunjungan di suatu *node*.

Setelah *deglet* bersifat permanent, maka *agent* tersebut dianggap sebagai suatu *netlet* (setelah suatu jaringan *agent*). *Netlet* adalah bagian dari infrastruktur suatu jaringan. Model jaringan tersebut dapat dikelola secara dinamis, karena *netlet* dapat mencari perubahan konfigurasi suatu jaringan. Beberapa *netlet* dapat ditugaskan untuk menjalankan perintah. Kecepatan suatu perubahan yang terdeteksi dapat dikendalikan melalui kepadatan *netlet*; semakin banyak *netlet*, semakin singkat proses pendeteksian. Suatu *netlet* dapat didefinisikan sebagai ruang lingkup jangkauan, namun tidak pernah meninggalkan sub-jaringan tertentu.



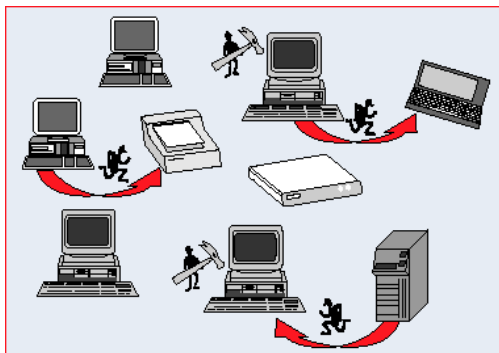
Gambar 2. Pemodelan Jaringan

Manajemen kesalahan

Manajemen kesalahan dibagi menjadi dua bagian:

1. Diagnosa Jaringan -- mempunyai prinsip yang sama seperti pemodelan jaringan yang digunakan untuk mendiagnosa kesalahan jaringan. Sebagai contoh, *deglet* dan *neglet* mempunyai manfaat masing-masing untuk menjalankan suatu kegiatan pada peralatan jaringan. Suatu *netlet* aktif, dapat digunakan untuk mencari suatu masalah secara otomatis yang kemudian dengan segera melakukan *recovery* jika memungkinkan. Oleh karenanya, manajer jaringan akan mendapatkan pesan, pemberitahuan dan informasi apabila jaringan

- membutuhkan keterlibatan manusia untuk *recovery* jaringan yang mengalami kesalahan.
2. Pengendalian jarak jauh elemen-elemen Heterogen.— *Mobile Agent* memerlukan interaksi dengan titik *hosting* melalui antarmuka yang menyediakan penanganan, akses taklangsung ke sumber daya *host* dan beberapa layanan lainnya.



Gambar 3. Interaksi *mobile agent*

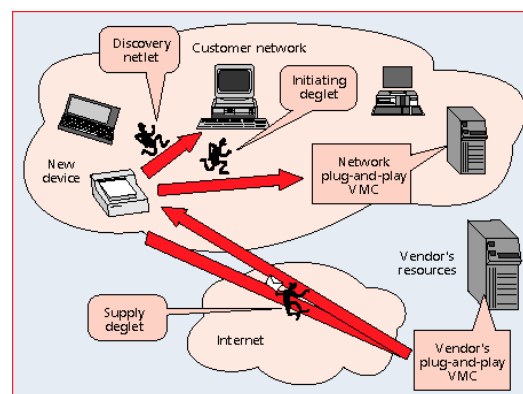
Manajemen Konfigurasi

Pensuplaian layanan –pensuplaian layanan dalam jaringan telekomunikasi merupakan proses yang kompleks, yang biasanya melibatkan beberapa pihak. *Mobile agent* membantu dalam hal *streamline* suatu proses. Hal tersebut dikenal sebagai aktivitas berelasi *agent* yang merupakan suatu organisasi kerja menuju standar pensuplaian layanan, yakni, Konsorsium Arsitektur Jaringan Informasi Telekomunikasi (*TINA-C*). Sebagai contoh Pensuplaian pada *Permanent Virtual Circuits (PVCs)* di jaringan *ATM*.

Sistem berbasis *mobile agent* dapat menangani perintah yang serupa secara otomatis [11]. Permintaan untuk mengaktifkan suatu *PVC* dilakukan oleh *deglet*, yang mengkoordinasikan proses keseluruhan. Hal tersebut memerlukan tambahan *deglet* untuk menjalankan perintah-perintah parsial lainnya. Dengan menggunakan *deglet* ini, semua data yang diperlukan akan dipertukarkan oleh *endpoint* dengan menggunakan pensuplaian yang terhubung dengan *Virtual Managed Components (VMCs)*. *Deglet-deglet* tersebut berkomunikasi dengan *VMCs* dengan menggunakan ontologi khusus yang menggeneralisasi pengetahuan /informasi untuk mengaktifkan koneksi silang dan *PVCs* yang dilakukan oleh vendor. Bagian data penting tersebut akan dibawa menuju lokasi yang jauh; sebagai contoh, situs web vendor. Kemudian, *deglet* lainnya akan melakukan hubungan dengan *VMCs* di awan operator (*operator's clouds*). Langkah terbaik yaitu dengan memilih dan mengurutkan pertukaran informasi-informasi penting yang lengkap untuk proses pengaktifan. Pada saat itu, pihak yang meminta akan mendapatkan pesan bahwa perintah telah selesai dijalankan.

Pensuplaian komponen – Konfigurasi suatu alat membutuhkan sejumlah atribut dalam jaringan dan alat tersebut harus diset pada tempat dimana komponen *software* di *install*. Sebagai contoh, suatu printer membutuhkan *driver* untuk digunakan pada satu stasiun kerja. Pada saat itu, manager jaringan harus menjalankan perintah-perintah yang dibutuhkan secara manual. *Mobil agent* dapat digunakan untuk mengimplemantasikan komponen jaringan *plug-and-play*. Sebagai gambaran, jika kita ingin meng-install printer jaringan, yang akan melibatkan stasiun-stasiun kerja yang mempunyai spesifikasi dan konfigurasi yang berbeda. Sebagai contoh, *driver* yang diperlukan untuk stasiun kerja antar Macintos dan Unix pastinya berbeda, begitu pula antara Windows NT dengan OS/2, Linux dengan Windows, dan lain-lain. Konfigurasi tersebut merupakan masalah yang harus dipecahkan.

Dengan *mobile agent* pensuplaian komponen dilakukan untuk mengatasi perbedaan konfigurasi yang berbeda. *Mobile agent* melakukan komunikasi antar stasiun kerja melalui jaringan, sejumlah *netlet* dan *deglet* dikirim untuk mencari alat-alat jaringan yang membutuhkan *driver* suatu printer. Kemudian, halaman web pembuat printer akan dihubungi, kemudian akan dilakukan *pen-downlad-an* versi terbaru *driver* printer yang dibutuhkan. Skema pensuplaian komponen diilustrasikan pada gambar 6.



Gambar 6. Componen Provision

Manajemen Kinerja

Beberapa aspek pengukuran kinerja jaringan cukup sulit untuk dilakukan apabila server tersentralisasi digunakan. Namun dengan *mobile agent*, pengukuran kinerja jaringan tidak lah terlalu sulit untuk dilakukan, karena dapat diukur secara lokal-per-lokal dan *mobile agent* tidak mengkonsumsi sumber daya lokal. Dalam kasus pengukuran kinerja ada dua teknologi yang dapat dilakukan yaitu, *hot-swapping* dan migrasi server.

Dalam konteks kinerja jaringan, pengukuran kinerja pada *mobile agent* salah satunya dengan mengukur kepadatan *agent-agent* yang digunakan,

semakin banyak *agent* yang terlibat, menyebabkan kepadatan semakin besar dan tidak mustahil berdampak pada penurunan kinerja jaringan. Pengoptimalan kinerja *agent* merupakan hal yang penting dalam kinerja jaringan. Untuk mengoptimalkan kinerja *agent* diperlukannya suatu pemberian informasi dan perintah-perintah yang jelas bagi *agent* dalam melaksanakan tugasnya, yang dilakukan oleh pihak yang berwenang (yakni operator manusia dan aplikasi sistem).

Manajemen Keamanan

Plug-and-Play Jaringan [2] – adalah suatu jaringan yang secara otomatis dapat mengkonfigurasi dirinya sendiri untuk mengakomodasi kebutuhan pengguna dan kebutuhan komponen. Sebagai contoh, suatu *applet* digunakan untuk mengadaptasi kehadiran data jaringan terhadap profil suatu alat dan profile penggunaannya.

Plug-and-Play Jaringan dapat mendeteksi masalah yang dapat mempengaruhi integritas dalam men-deteriorasi kualitas pelayanan atau masalah keamanan. Jika suatu masalah terdeteksi, maka jaringan akan melakukan perbaikannya sendiri secara otomatis .

4. Kesimpulan

Konsep klasik manajemen jaringan nampak tidak memadai lagi seiring dengan bertambahnya ukuran, manajemen kompleksitas, kebutuhan layanan yang semakin tinggi ketuhan infrastruktur yang semakin bertambah. Untuk itu perlu adanya perubahan paradigma ataupun penyempurnaan, sebagai alternative baru, yaitu manajemen jaringan berbasis *mobil agent* dimana menawarkan sejumlah keunggulan yang tidak dijumpai dari penggunaan konsep sebelumnya.

Manajemen jaringan berbasis *mobil agent* dalam pengembangannya tetap mempertimbangkan model dasar manajemen OSI, yaitu: manajemen kesalahan, manajemen akuntansi, manajemen konfigurasi, manajemen kinerja dan manajemen keamanan.

Pengembangan terakhir dalam penggunaannya di Internet mengindikasikan bahwa *mobil agent* banyak digunakan untuk melakukan perintah-perintah yang beragam di banyak aplikasi. Dari sekian banyak berita yang cukup menggembirakan tersebut, pengembangan konsep *mobil agent* memerlukan penelitian dan kajian yang lebih mendalam dan menyeluruh lagi mulai penetapan standarisasi, kerjasama dengan para vendor hingga pengembangan aplikasi yang sesuai dengan kebutuhan pengguna.

Dan yang perlu perhatikan semua pihak khususnya para peneliti adalah mengenai pengembangan *mobil agent* yang terkoneksi dengan jaringan Internet untuk perancangan system yang lebih baik. Sebagai informasi tambahan,

pengembangan terbaru yaitu software Jini yang merupakan perpaduan teknologi yang berbasis pada *remote object* (seperti CORBA), *mobile code and agent*.

Daftar Pustaka

- [1] Mobile Agent Facility Specification, OMG TC Document cf/xx-x-xx, June 2, 1997.
- [2] Yemini, Y., *The OSI Network Management Model*, *IEEE Commun. Mag.*, May 1993, pp. 20-29.
- [3] Green, S. et al., *Software Agents: A review*, Technical Report, Department of Computer Science, Trinity College, Dublin, Ireland.
- [4] Mobile Code Bibliography. <http://www.cnri.reston.va.us/home/koe/bib/mobile-abs.bib.html>
- [5] Baldi, M., Gai, S. and Picco, G. P., *Exploiting Code Mobility in Decentralized and Flexible Network Management*, First Int'l Workshop on Mobile Agents Mobile Agents '97, Berlin, Germany, April 7-8, 1997.
- [6] Vinoski, S., *CORBA overview: CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments*, *IEEE Commun. Mag.*, vol. 14, no. 2, February 1997.
- [7] <http://java.sun.com/products/jini>
- [8] Schramm, C., Bieszczad, A. and Pagurek, B., *Application-Oriented Network Modeling with Mobile Agents*. Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS '98), New Orleans, Louisiana, Feb. 1998.
- [9] Bieszczad, A. and Pagurek, B., *Network Management Application-Oriented Taxonomy of Mobile Code*, Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS '98), New Orleans, Louisiana, Feb. 15-20, 1998.
- [10] Susilo, G., Bieszczad, A. and Pagurek, B., *Infrastructure for Advanced Network Management based on Mobile Code*. Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS '98), New Orleans, Louisiana, Feb. 15-20, 1998.
- [11] Pagurek B. et al., *Network Configuration Management In Heterogeneous ATM Environments*, Proc. of the Int'l Workshop on Agents in Telecommunications Applications (IATA '98), AgentWorld'98, Paris, France, July 4-7, 1998.