

Implementasi Software Mobile Agent Berbasis Java

Oleh : Muhamad Ali,
Dosen Jurusan PT Elektro Fakultas Teknik
Universitas Negeri Yogyakarta

Abstrak

Penelitian ini bertujuan untuk menjelaskan definisi software mobile agent, kelebihan-kelebihannya, implementasi dan pengembangannya dalam aplikasi berbasis web dengan bahasa pemrograman Java.

Penelitian ini menggunakan metode pengembangan software secara umum yang terdiri dari 4 tahap. Pertama, analisis berorientasi objek untuk menentukan kebutuhan sistem. Kedua, pemodelan sistem yang merupakan gambaran urutan proses dan hubungan antara proses secara mendetail dalam program. Ketiga, implementasi yaitu tahap menterjemahkan modul-modul hasil desain dengan menggunakan bahasa pemrograman ke dalam bentuk aplikasi software mobile agent. Keempat, pengujian, software mobile agent diuji pada jaringan komputer (LAN) yang terdiri dari tiga komputer dengan sistem operasi yang berbeda.

Hasil penelitian menunjukkan bahwa software agent dapat dikembangkan dengan bahasa pemrograman Java. Berdasarkan pengujian, sistem mobile agent dapat bekerja dengan baik pada sistem operasi yang berbeda.

Kata Kunci: mobile agent, java, multipatform, agents server page

Pendahuluan

Agen merupakan suatu teknologi baru dalam dunia rekayasa software. Kemampuan agen yang otonom, personal, mobile dan mampu bekerja sama secara terus menerus membuat para pengembang software memanfaatkan teknologi ini untuk aplikasi yang mempunyai lingkungan yang memiliki jenis informasi yang beragam dan tersebar di banyak lokasi (Langeet al, 1999).

Software agent dapat dikembangkan dengan bahasa pemrograman yang dapat berjalan di berbagai platform sistem operasi agar dapat beroperasi dalam jaringan berbasis web. Bahasa pemrograman yang sering digunakan adalah Java, yang merupakan bahasa pemrograman sederhana, berorientasi objek, multithreaded, garbage collected, aman, robust, berarsitektur netral, portable, berkinerja tinggi, dan dinamis.

Bahasa Pemrograman Java memiliki keunggulan yaitu (Qusay, 2000)

- *Platform-independence*, Java dapat dioperasikan dalam beragam jaringan yang tidak tergantung pada sistem operasi (Windows, Linux, Mac, Solaris, dan lain-lain)
- *Secure execution*, Java memiliki keamanan yang baik bila dijalankan oleh host sehingga sangat cocok diaplikasikan dalam intranet dan internet.
- *Dynamic class loading*, memungkinkan JVM (*Java Virtual Machine*) memuat dan mendefinisikan class-class saat program dijalankan serta loading melalui jaringan.
- *Object serialization*, salah satu fitur mobile agent adalah serialisasi dan deserialisasi yang disediakan oleh Java
- *Multithread programming*, Java mendukung proses multithreading dan juga mendukung proses sinkronisasi untuk interaksi antar agen.
- *Reflection*, kode Java mengetahui informasi tentang class yang di-load, fields, methods, dan constructors.

Software Agent

Untuk memahami konsep *software agent* dengan lebih jelas perlu diketahui definisinya. Ada beberapa definisi *software agent* menurut para ahli yang didasarkan pada kebutuhan dan kepentingannya. Berikut ini beberapa definisi *software agent* menurut para ahli, antara lain Graesser dalam (Jening et. al, 1998) mengemukakan *agent* merupakan sebuah sistem yang bekerja pada lingkungan tertentu dengan tugas untuk melakukan penginderaan terhadap perubahan lingkungan sesuai dengan yang diperintahkan. Sedangkan (Sycara et. al, 1998) mengemukakan *agent* adalah sistem komputer yang mampu beradaptasi dalam lingkungan tertentu dan bertugas sesuai dengan yang diprogramkan.

Pakar lain dalam (Jening, et al, 1998) mendefinisikan *agent* sebagai berikut : *Agent* merupakan sistem yang bekerja secara mandiri dan berorientasi pada lingkungan, *agent* merupakan sensor dan aktuator terhadap lingkungan, *agent* merupakan sistem yang mampu menganalisis keadaan lingkungan, bekerja sesuai dengan tujuan, *agent* merupakan interpretasi kondisi lingkungan yang dinamis *Agent* merupakan objek yang mempunyai tujuan dan bekerja secara mandiri. *Agent* merupakan objek yang dapat berpindah dalam lingkungan, *agent* merupakan program yang dapat berinteraksi dengan lingkungan seperti sistem operasi, internet, basis data guna mencapai tujuan tertentu.

Dari penjelasan di atas dapat ditarik suatu definisi *software agent* yang merupakan rangkuman dari beberapa definisi yang telah dikembangkan para pakar. Definisi *software agent* yang akan dipakai dalam penelitian ini dapat dilihat dalam dua perspektif, yaitu perspektif *user* dan perspektif sistem. Dalam perspektif *user software agent* merupakan sebuah *software* yang bertindak selaku perantara/agen atau *broker* bagi *user* yang memungkinkan *user* untuk mendelegasikan tugas kepadanya serta melakukan pekerjaan seperti yang diperintahkannya. Sedangkan dalam perspektif sistem, agen dapat mengenali lingkungan kerjanya dan memiliki sifat-sifat keagenan. Sifat-sifat yang harus dimiliki agen adalah sebagai berikut (Lange et. al., 1998) :

1. Menyatu dengan lingkungannya
2. Memiliki sifat-sifat yang dimandatkan antara lain:
 - *Reactive* – dapat merasakan perubahan pada lingkungannya dan bertindak sesuai dengan perubahan lingkungan tersebut.
 - *Autonomous* – memiliki sistem kontrol terhadap tindakannya sendiri.
 - *Goal-driven* – bersifat *pro-active* untuk mencapai tujuan
 - *Temporally continuous* – dapat melanjutkan proses eksekusi di tempat lain.
3. Memiliki salah satu atau lebih sifat-sifat berikut
 - *Communicative* – dapat berkomunikasi dengan agen lain
 - *Mobile* – dapat berpindah dari satu host ke host lain
 - *Learning* – beradaptasi sesuai dengan pengalaman sebelumnya
 - *Believable* – dapat dipercaya pada tingkat *end-user*, agen tidak akan merusak data *user*.

1. Mobile Agent

Mobile agent ialah agen yang mempunyai kemampuan berpindah-pindah dalam jaringan (contohnya pada jaringan *World Wide Web*), berinteraksi dengan *host-host* asing, mengumpulkan informasi berdasarkan pengguna, dan mengembalikannya ke pengirim setelah melakukan tugasnya. *Mobile agent* ini diimplementasikan oleh *remote program*. Hal-hal yang berkenaan dengan pemrograman *remote* juga dapat diterapkan pada *mobile agent* seperti (Lange et. al, 1998) :

- Penamaan program - memberikan nama kepada agen untuk membedakannya dengan agen yang lain.
- Autentifikasi program - autentifikasi implementor dari program agen.

- Migrasi program - memindahkan program dari satu mesin ke mesin lainnya
- Keamanan program - memastikan program tidak merusak mesin pengeksekusinya

Keuntungan *Mobile Agent*

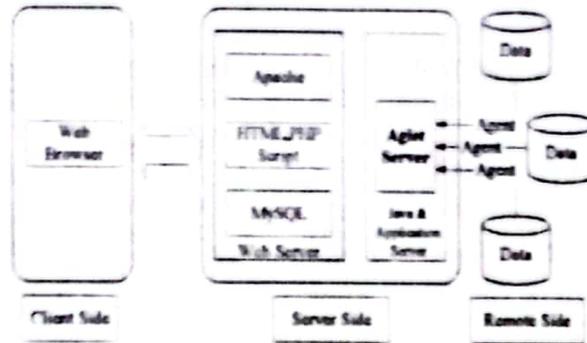
Dibandingkan dengan teknologi agen statis, *mobile agent* mempunyai beberapa keuntungan yaitu :

- Mengurangi Beban Jaringan.**
Pada sistem terdistribusi komunikasi antar subsistem sangat tergantung pada protokol komunikasi yang melibatkan banyak interaksi untuk menyelesaikan tugas yang diberikan. Hal ini menyebabkan, trafik jaringan tinggi. Dengan *mobile agent* dapat dimungkinkan untuk mengemat suatu aplikasi, mengirimkannya ke *host* tujuan dan kemudian interaksi dapat terjadi secara lokal.
- Efisiensi sumber daya.**
Konsumsi sumber daya (CPU dan memori) dapat dihemat, sebab *mobile agent* bekerja sesuai dengan tugas yang diberikan yaitu pada satu *node* pada satu waktu. *Node* yang lain tidak menjalankan agen sampai *node* tersebut memerlukannya.
- Menanggulangi *latency* jaringan.**
Sistem *real-time* yang kritis perlu tanggap terhadap perubahan lingkungannya secara *real-time*. Keterlambatan tanggapan yang diakibatkan oleh masalah jaringan harus dihindari. *Mobile agent* menawarkan suatu pemecahan dengan mengirimkan agen ke tujuan dan dieksekusi secara lokal.
- Eksekusi secara *Asynchronous dan Autonomous*.**
Untuk menjaga koneksi dengan sistem lainnya diperlukan sarana komunikasi yang baik. Sayangnya dalam kenyataan seringkali banyak ditemukan jaringan yang mahal dan mudah putus, sehingga untuk mempertahankan konektivitas menjadi tidak *feasible*, baik secara ekonomis maupun teknis. Dengan *mobile agent* hal ini dapat diminimasi karena *mobile agent* mampu bekerja secara mandiri dan dapat beroperasi secara *asynchronous dan autonomous*.
- Adaptasi secara dinamis.**
Mobile agent dapat mendeteksi adanya perubahan di lingkungannya dan dapat bereaksi secara *autonomous* melakukan perubahan.
- Andal dan toleran terhadap kesalahan.**
Kemampuan *mobile agent* untuk beraksi secara dinamis pada situasi dan keadaan yang tak menguntungkan menjadikan *mobile agent* mudah untuk membuat sistem terdistribusi yang andal dan toleran terhadap kesalahan.
- Multiplatform*.**
Dalam komputasi berbasis web sangat dimungkinkan penggunaan berbagai sistem yang berbeda baik pada sisi *hardware* maupun *software*. *Mobile agent* tidak tergantung pada komputer dan jaringan, tetapi hanya tergantung pada lingkungannya. Sebagai contoh, *mobile agent* Java dapat ditujukan ke segala sistem yang mempunyai JVM (*Java Virtual Machine*) dan dapat beroperasi pada berbagai sistem operasi.

Perancangan Sistem *Mobile Agent*

Sistem *mobile agent* yang akan dibangun adalah *agent* untuk melakukan *monitoring* dan *updating* data stok di sebuah distributor dalam jaringan berbasis web. Secara lengkap sistem yang akan dibangun pada penelitian ini dapat dijelaskan seperti pada Gambar 1. Masing-masing komputer mempunyai tugas fungsi sebagai berikut :

- ↳ Web server
- ↳ Server basis data
- ↳ Klien



Gambar 1 Rancangan sistem

1. Analisis Berorientasi Objek (*Object Oriented Analysis*)

Analisis dimaksudkan untuk melihat dan mendeskripsikan kebutuhan-kebutuhan sistem dan membuat suatu model yang menentukan kelas-kelas dalam sistem. Proses analisis sistem dalam penelitian ini dilakukan dengan pendekatan berbasis objek untuk menggambarkan keadaan sistem. Analisis sistem dilakukan dengan langkah-langkah sebagai berikut :

- Identifikasi aktor
- Pengembangan *uses case*
- Pengembangan diagram interaksi
- Pengembangan kelas

a. Identifikasi Aktor

Aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem, atau dengan kata lain siapa atau apa yang menggunakan sistem (Booch et. al, 1998). Dalam interaksinya dengan sistem, aktor melakukan interaksi berupa mengirim ataupun menerima informasi dari sistem. Dari penjelasan ini dapat dikatakan bahwa aktor dapat bersifat aktif dengan melakukan inisiasi *use case* atau dapat juga bersifat pasif yang tidak menginisiasi *use case*. Aktor-aktor dalam sistem yang dikembangkan adalah sebagai berikut :

- ↳ Komputer Pemantau
Aktor dalam komputer pemantau adalah administrator program pengguna komputer yang dapat melimpahkan tugasnya kepada *software statis agent* "AgenPengawas" dan *software mobile agent* "PengawasMobile" untuk melakukan *monitoring* data di komputer yang akan dipantau.
- ↳ Komputer yang Dipantau
Aktor dalam Komputer yang Dipantau adalah administrator yang bertugas mengaktifkan Aglets Server agar komunikasi antara Komputer yang pemantau dan Komputer yang Dipantau dapat berjalan dengan baik.
- ↳ Pengguna
Pengguna menginisiasi *use case* dengan melakukan transaksi pembelian ke distributor melalui internet

b. Pengembangan Use Case

Setelah ditentukan aktor, selanjutnya dideskripsikan bagaimana interaksi mereka dengan elemen sistem lainnya. Masing-masing aktor perlu dideskripsikan tugas dan fungsinya dalam

sistem. Dalam pengembangan *use case* pada penelitian ini terdapat 3 macam pengembangan yaitu *monitoring*, *upkating*, dan transaksi *on-line*

1). *Monitoring*

- ✓ Admin sistem pemantau masuk ke dalam sistem
- ✓ Mengaktifkan "Aglets Server"
- ✓ Mengcreate *static agent* "AgenPengawas"
- ✓ Melakukan pengaturan tugas *mobile agent* "PengawasMobile"
- ✓ Menentukan *host* tujuan
- ✓ Melakukan pengaturan parameter
- ✓ Mengirim *mobile agent* "PengawasMobile" ke *host* tujuan
- ✓ *Static agent* "PengawasMobile" Memonitor data di komputer tujuan
- ✓ *Mobile agent* "PengawasMobile" melaporkan setiap terjadi perubahan basis data di basis data server yang dimonitor
- ✓ *Mobile agent* "PengawasMobile" mengcreate "messenger agent"
- ✓ "Messenger agent" melaporkan perubahan data ke *static agent* "Agen Pengawas" di server komputer pemantau
- ✓ *static agent* "Agen Pengawas" mencatat data yang berubah
- ✓ *static agent* "Agen Pengawas" melakukan *upkate* basis data di server komputer pemantau

2). *Updating*

- ✓ Admin Komputer Pemantau Mengaktifkan program *web browser*
- ✓ Masuk ke menu *upkating*
- ✓ *Login* ke sistem
- ✓ Masuk ke menu pencarian dengan agen
- ✓ Melakukan pencarian data dengan mesin pencari
- ✓ Mesin pencari mencari data *upkate* terbaru
- ✓ Admin Komputer Pemantau melakukan *upkate* basis data di server perusahaan manufaktur

3). Transaksi *on-line* di Komputer yang Dipantau

- ✓ User mengaktifkan program *web browser*
- ✓ Masuk ke alamat URL server tujuan (distributor)
- ✓ *Login* untuk verifikasi
- ✓ Masuk ke menu katalog produk yang disediakan
- ✓ Memilih produk yang ingin dibeli
- ✓ Melakukan pembayaran via kartu kredit
- ✓ Sistem melakukan verifikasi kartu kredit
- ✓ Menjalankan transaksi
- ✓ Melaporkan kepada pelanggan perihal waktu pengiriman

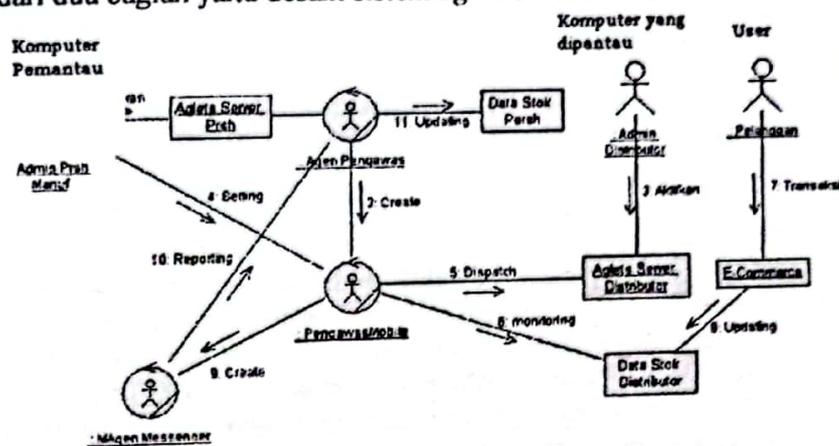
c. Diagram Interaksi

Dalam sistem berorientasi objek, objek-objek dalam sistem saling berkomunikasi dengan mengirimkan pesan atau *message*. Interaksi dan komunikasi diantara objek-objek dalam sistem ini merupakan perilaku sistem dinamis. Aspek-aspek dinamis ini dalam *Unified Modelling Language* (UML) dimodelkan dengan diagram interaksi (Booch et. al, 1998). Interaksi antar objek dalam sistem yang dibangun akan dijelaskan dengan diagram interaksi yang meliputi diagram kolaborasi sistem agen dan sistem transaksi *on line*.

2. Desain Sistem

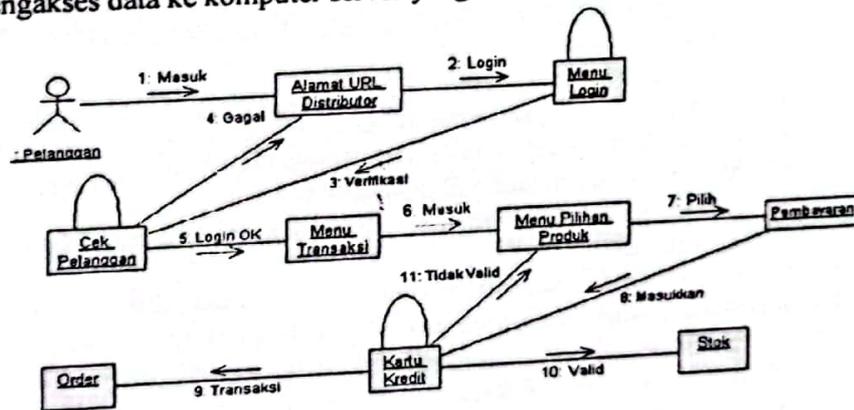
Desain pada sistem ini menggunakan desain berbasis objek dengan menggunakan pemodelan grafis yang menggambarkan hubungan antar objek dalam sistem. Hubungan

antar objek digambarkan dalam bentuk diagram kolaborasi seperti pada gambar 2. Desain sistem terdiri dari dua bagian yaitu desain sistem *agent* dan desain sistem transaksi *on line*.



Gambar 2. Diagram kolaborasi sistem

Desain sistem transaksi *on line* dapat dijelaskan pada gambar di 3. *User* melakukan transaksi *on line* untuk mengakses data ke komputer server yang akan dimonitor oleh *mobile agent*.



Gambar 3. Diagram kolaborasi transaksi *on-line*

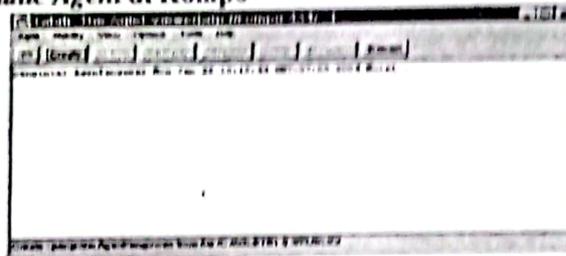
3. Implementasi Sistem

Sistem dibangun dengan bahasa pemrograman Java yang terdiri dari *static agent* yaitu "Agen Pengawas" dan *mobile agent* yaitu "PengawasMobile" dan "messenger". Sistem bekerja dalam lingkungan *aglets server page (ATP://)*. *Static agent* "Agen pengawas" diciptakan di komputer pemantau. *Static agent* "Agen pengawas" akan menciptakan *mobile agent* "PengawasMobile" yang akan berpindah dari komputer pemantau ke komputer yang dipantau. *Mobile agent* "PengawasMobile" akan bekerja jika terjadi perubahan data di komputer yang dipantau. Perubahan data di komputer yang dipantau akan dilakukan dengan simulasi transaksi *on-line*.

4. Pengujian Sistem

Pengujian sistem dilakukan dengan mengujicobakan sistem *agent* untuk bekerja dalam jaringan *local area network (LAN)* yang terdiri dari Komp0 sebagai komputer pemantau, Komp1 sebagai komputer yang dipantau dan Komp2 sebagai komputer klien yang akan mengakses data ke Komp1 melalui transaksi *on line*.

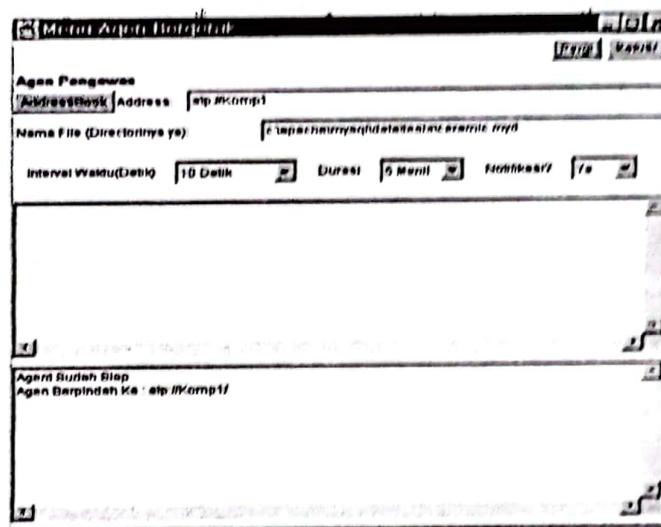
a). Pengujian *Static Agent* di Komp0



Gambar 4. Agen Pengawas pada saat di create di komputer pemantau

Gambar di atas menunjukkan proses pengaktifan *static agent* "Agen Pengawas" pada komputer pemantau (komp0). *Agent* berhasil diaktifkan di Komp0. Setelah *static agent* "Agen Pengawas" berhasil diciptakan maka langkah selanjutnya adalah menguji pengaktifan *mobile agent* "PengawasMobile".

b). Pengujian *Mobile Agent* di Komp1



Gambar 5. Pengiriman Agen PengawasMobile ke komputer tujuan
Gambar 5. menunjukkan proses pengaktifan *mobile agent* "PengawasMobile" pada komputer pemantau (komp0) dan dikirim ke komputer yang dipantau (komp1). Dari gambar tersebut terlihat *mobile agent* "PengawasMobile" berhasil dikirim dari Komp0 ke Komp1

3. Sistem agen yang dibangun meliputi Agen Pengawas, Agen Pengawas mobile, agen messenger dan agen updating data.
4. Masing-masing agen dapat bekerja sesuai dengan tugas yang diberikan kepadanya dan dapat berkomunikasi dengan baik.

Daftar Pustaka

1. *Booch, G., Rumbaugh, J., Jacobson, I., (1998), "The Unified Modelling Language User Guide", Addison-Wesley, Massachusetts.*
2. *Lange, Danny B. and Mitsuru Oshima, . (1998), "Programming and Deploying Java Mobile Agents with Aglets", Addison-Wesley.*
3. *Lange, Danny and M. Oshima, (1999), "Seven good reasons for mobile agents". Communications of the ACM 42 p. 88-89.*
4. *Mahmoud, H Qusay, (2000) "Distributed Programming With Java", Manning Pub...*
5. *Nicholas R. Jennings and Michael J. Wooldridge, (1998), "Agent Technology: Foundations, Applications and Markets", Springer Verlag.*
6. *Nicholas R. Jennings, Katia Sycara, Michael Wooldridge, (1998), "A Roadmap of Agent Research and Development," Journal of Autonomous Agents and Multi-Agent Systems.*

Riwayat Penulis

Muhamad Ali, ST, MT lahir di Pekalongan, 27 November 1974. Penulis menamatkan pendidikan S₁ di Jurusan Teknik Elektro UGM pada tahun 1997 dalam waktu 4 tahun 10 bulan. Pendidikan S₂ diselesaikan di Teknik Industri ITB dengan Konsentrasi pada Pengembangan Sistem Informasi Enterprise. Pekerjaan sekarang sebagai dosen tetap di Jurusan Pendidikan Teknik Elektro Fakultas Teknik Universitas Negeri Yogyakarta (UNY) sejak tahun 2000. Selain mengajar dan penelitian, penulis juga aktif menulis artikel bidang Teknik dan Pendidikan Elektro yang dipublikasikan baik internal UNY maupun ke luar.