

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK BERBASIS MOBILE AGENT UNTUK PENCARIAN BUKU

Sudjadi, Maman Somantri, Scifo Anggi Yudanto, and Hanung Prasetyo

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

Abstrak

The development of information technology today is growing very fast. The software not only required in running a fast execution but also efficient in the utilization of network bandwidth so that traffic does not become heavy. With the swift network-based computing devices into everyday life, the higher the demand to take advantage of technology agent in the process of problem solving based mobility. This study purpose to implement technology on a mobile Agent software with a case study of a software used to perform a book search on multiple database servers. In software, the agent will move from one database server to another server database to find books that sought out and returned to the client computer and show the results obtained. In this research using Aglet 2.0.2 as the software used for build of such applications. Aglet is used to develop agents who will move from one computer to another computer. To view the search results using PHP and HTML language. Based on the results of tests have been done, the conclusion that application could be implemented by the method of Mobile Agents. Mobile Agents used consists of three kind of AgentOne which serves to make AgentTwo and send it to computer database server, AgentTwo which serves to find the books what are looking for and make AgentThree, and AgentThree which serves to bring the results and go back to the client computer. These three agents formed a collaboration of good programs so the system book search can work as expected.

Keywords : Agent, Mobile Agent, Aglets, Book Searching

1. Pendahuluan

1.1 Latar Belakang Masalah

Perkembangan teknologi dan informasi dewasa ini tumbuh sangat cepat. Perangkat lunak tidak hanya dituntut cepat dalam menjalankan suatu eksekusi tetapi juga efisien dalam pemanfaatan *bandwidth* sehingga trafik jaringan tidak menjadi berat. Dengan semakin derasnya intervensi perangkat komputer berbasis jaringan ke dalam kehidupan sehari-hari, semakin tinggi pula tuntutan untuk memanfaatkan teknologi agen berpindah dalam proses pemecahan masalah (*problem-solving*) berbasis mobilitas.

Berdasar pada pengertian di atas, dapat dikatakan bahwa agen berpindah pada dasarnya adalah objek berpindah. Agen berpindah direalisasikan dengan mengirimkan kode program dari sebuah objek dalam bahasa Java dari satu lokasi ke lokasi lain. Hal ini mirip dengan *applet*, tetapi agen berpindah tidak mengikuti paradigma *client-server* yang dipakai *applet*. *Mobile agent* merupakan suatu objek yang secara otomatis bergerak melalui suatu jaringan dari tempat yang satu ke tempat yang lain dengan kode serta status yang telah dieksekusi dan menampilkannya atas nama pengguna.

Menurut Lange dan Oshima ^[1] *Mobile Agent* mempunyai beberapa keuntungan yaitu :

1. Mengurangi trafik jaringan karena hanya menggunakan jaringan saat berpindah saja.
2. Mengatasi *network latency*, Sistem *real time* yang kritis, dimana latensi tak dibolehkan, *mobile agent* memberikan solusi yang baik, karena agen dapat berpindah dari pusat kontroler ke kontroler lokal dan dapat menjalankan program secara langsung di kontroler lokal.
3. *Asynchronous and autonomous*, Agen dapat ditugaskan dan dikirim ke jaringan.
4. Beradaptasi secara dinamis dengan lingkungan, *Mobile Agent* memiliki kemampuan beradaptasi dengan lingkungannya
5. Heterogen, Komputer jaringan pada dasarnya heterogen baik perangkat lunak maupun perangkat keras
6. *Robust and fault-tolerant*, Kemampuan bereaksi terhadap situasi tidak menguntungkan dan memiliki toleransi terhadap kesalahan

Dalam penelitian ini diambil studi kasus berupa suatu perangkat lunak yang digunakan untuk melakukan pencarian buku pada beberapa database *server*. Dalam perangkat lunak ini *agent* akan bergerak dari database *server* satu ke database *server* lainnya untuk mencari buku yang dicari dan kembali ke komputer *client* serta menunjukkan hasil yang didapatkan. Diharapkan dengan pembahasan ini dapat digunakan sebagai dasar tentang

konsep *mobile agent* dan implementasi dalam pembuatan suatu perangkat lunak ataupun penelitian tentang *mobile agent*.

1.2 Tujuan

Tujuan penelitian ini adalah untuk mengkaji :

1. Teknologi *Mobile Agent* dapat diaplikasikan ke dalam suatu perangkat lunak menggunakan bahasa pemrograman Java.
2. Dapat diketahui spesifikasi, kebutuhan, metode, ciri-ciri, kemampuan dan karakteristik teknologi *Mobile Agent* ini pada aplikasi ini.
3. Didapatkan informasi buku yang dicari dari berbagai database *server*.

1.3 Batasan Masalah

Agar pembahasan atau analisis tidak melebar dan terarah, maka permasalahan dibatasi pada :

1. Data yang digunakan dalam penelitian ini adalah data simulasi.
2. Proses pencarian hanya pada judul, pengarang, dan penerbit buku saja.
3. Penelitian ini ditekankan pada pembuatan, komunikasi dan migrasi *Mobile Agent* di dalam jaringan untuk mendapatkan informasi yang diinginkan
4. Penelitian tidak membahas mengenai keamanan (*security*) dari sistem *Mobile Agent* ini.
5. Jaringan hanya menggunakan alamat IP statis.
6. Sistem operasi yang digunakan adalah Linux Ubuntu.
7. Framework *Mobile Agent* yang digunakan adalah *aglets 2.0.2*, PHP untuk merancang situs dan *Apache Server* untuk *web hosting* dari jaringan lokal.
8. Sistem ini hanya bisa berjalan pada *host* yang memiliki *Java Virtual Machine*, *Aglets* dan *Apache Server*

2. Landasan Teori

2.1 Agent

Di dalam kamus *Webster's New World Dictionary* ^[2], *agent* didefinisikan sebagai :

A person or thing that acts or is capable of acting or is empowered to act, for another.

Dari definisi diatas, Somantri ^[3] menyimpulkan ada dua hal penting tentang agen, yaitu:

- a. *Agent* mempunyai kemampuan untuk melakukan suatu tugas/pekerjaan.
- b. *Agent* melakukan suatu tugas/pekerjaan dalam kapasitas untuk sesuatu, atau untuk orang lain.

1. Sejarah Agent

Menurut Nwana ^[4], konsep *agent* sudah dikenal lama dalam bidang kecerdasan buatan (*Artificial Intelligence-AI*), tepatnya dikenalkan oleh seorang peneliti bernama Carl Hewitt ^[5] dengan *concurrent actor model*-nya pada

tahun 1977. Dalam modelnya Hewitt mengemukakan teori tentang suatu objek yang yang dia sebut *actor*, yang mempunyai karakteristik menguasai dirinya sendiri, interaktif, dan bisa menanggapi pesan yang datang dari lain objek sejenis.

2. Mobile Agent

Menurut Lange ^[12] *Mobile Agent* adalah agen yang aktif dan dapat bergerak menuju komputer lain, atau menjelajahi jaringan untuk menjalankan tugasnya. *Mobile Agent* sering digunakan untuk mengumpulkan data, informasi atau suatu perubahan.

Pada *Aglets platform* digunakan *Agent Transfer Protokol* (ATP) untuk melakukan transfer *agent*. ATP merupakan suatu protocol yang didesain untuk membawa *agent* secara bebas pada suatu sistem. Menurut Oshima dkk ^[14] ATP memiliki metode standar seperti ditunjukkan pada Gambar 1 yaitu :

1. *Dispatch* : proses memindahkan *agent* dari satu *host* ke *host* lainnya dalam suatu jaringan.
2. *Retract* : proses mengembalikan *agent* dari *host* tujuan ke *host* pengirim.
3. *Fetch* : suatu metode yang digunakan untuk mendapatkan informasi.
4. *Message* : suatu pesan yang dikirimkan melalui suatu jaringan agar dapat berpindah dari satu protocol ke protocol yang lain.
5. *Response* : suatu jawaban penerima pesan ke pengirim pesan.



Gambar 1 Proses *Agent Transfer Protocol* (ATP)

2.2 AGLET

IBM *Aglets Workbench* (*Aglets*) dikembangkan oleh Danny B. Langedan Mitsuru Oshima dari IBM tokyo Research laboratory pada 1996. Menurut Lange ^[15], *Aglets* adalah objek Java yang dapat bergerak dari satu *host* ke *host* lain dalam suatu jaringan. Oshima dkk ^[14] juga mengatakan bahwa *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 *state* dari semua objek yang membawanya. Sebuah mekanisme keamanan yang *built-in* akan mengamankan *host* dari *untrusted Aglet*.

Istilah *Aglet* sesungguhnya adalah kombinasi dari kata agen dan applet. Perbedaan dengan applet adalah *Aglet* juga membawa serta *state*, dan memiliki *itinerary* (rencana perjalanan).

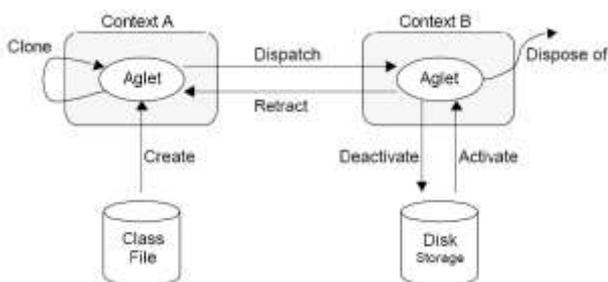
Tahiti adalah suatu program aplikasi yang bekerja sebagai *server Aglet*. Aplikasi tahiti merupakan suatu paket dengan ASDK. Beberapa *server* (tahiti) dapat dijalankan dalam satu komputer dengan cara memberikan nomor port yang berbeda. Tahiti menyediakan *pengguna interface* untuk pemantauan, penciptaan,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

Menurut Oshima dkk^[14] elemen dasar dari *Aglet* yaitu :

1. *Aglet* : Sebuah objek java yang dapat dipindahkan dengan membawa status dan informasi. Ini bersifat otonom dan reaktif.
2. *Proxy* : bagian dari *Aglet* yang berfungsi menyediakan lokasi *Aglet* secara transparan, sehingga dapat menyembunyikan lokasi *Aglet* yang sebenarnya. *Aglet* dan alamat dari suatu *Aglet* dapat dipisahkan sehingga *proxy* local di tiap terminal dapat menyembunyikan alamat dari suatu *Aglet*.
3. *Context* : wilayah kerja dari *Aglet*, objek yang diperlukan untuk mengatur dan mengelola jalannya *Aglet* pada suatu lingkungan.
4. *Identifier* : sebuah variabel global yang digunakan dalam *Aglet* dan hanya berlaku saat *Aglet* aktif.

1. Siklus Hidup Aglet

Setelah diciptakan, *Aglet* dapat melakukan operasi-operasi dasar seperti yang dijelaskan dibawah ini. Menurut Oshima dkk^[14] siklus hidup suatu *Aglet* seperti digambarkan pada Gambar 2



Gambar 2 Siklus hidup Aglet

1. *Create*: penciptaan sebuah *Aglet*. *Create* terjadi di dalam *context*. *Aglet* yang baru diberi sebuah pengenalan, dimasukkan ke dalam *context* dan diinisialisasi. *Aglet* mulai eksekusi segera setelah inisialisasi sukses.
2. *Clone*: proses penggandaan sebuah *Aglet*. *Clone* menghasilkan turunan (*copy*) yang hampir identik dengan *Aglet* yang asli didalam *context* yang sama. Perbedaannya hanya terletak pada *identifier* yang

diberikan dan eksekusi *Aglet* baru hasil cloning dimulai dari awal (*restart*). Catatan bahwa thread eksekusi tidak di *-clone*.

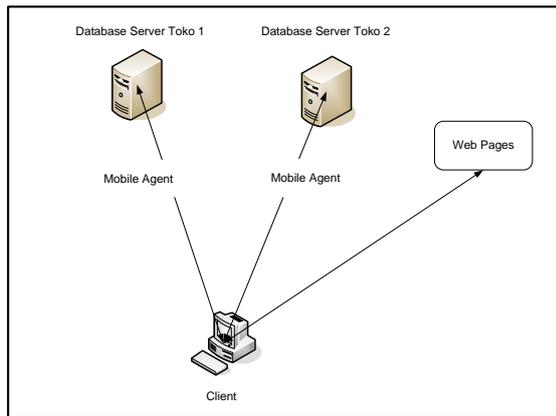
3. *Dispatch*: pemindahan sebuah *Aglet* dari satu *context* ke *context* yang lain. *Dispatch* akan memindahkan *Aglet* dari *context* yang sedang berlangsung, masuk ke *context* tujuan dan kemudian memulai awal eksekusinya.
4. *Retract*: proses untuk menarik kembali *Aglet* dari *context* yang sedang berlangsung dan masuk ke *context* yang melakukan permintaan kembali.
5. *Activate*: kemampuan untuk mengembalikan *Aglet* ke dalam *context*.
6. *Deactivate*: kemampuan untuk menghentikan sementara jalannya eksekusi *Aglet* dan menyimpan *state Aglet* dalam penyimpanan sekunder.
7. *Dispose*: proses untuk menghentikan jalannya eksekusi *Aglet* yang sedang berlangsung dan mengeluarkan *Aglet* dari *context* yang sedang berlangsung.

2.3 MySQL

MySQL adalah perangkat lunak Sistem Manajemen Basisdata (DBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public lisenche*). Suatu relasional basisdata menyimpan data dalam table table terpisah. MySQL merupakan turunan salah satu konsep utama dalam basisdata sejak lama yaitu SQL (*Structured Query Language*). MySQL menggunakan standar SQL (*Structured Query Language*), yaitu bahasa standar yang paling banyak digunakan untuk mengakses basisdata dan SQL dirancang khusus untuk berkomunikasi dengan basisdata.

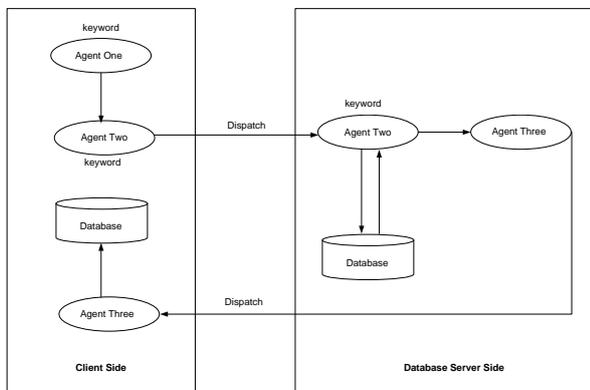
3. Perancangan Perangkat Lunak

Penelitian ini dirancang untuk mengimplementasikan konsep *mobile agent* ke dalam sistem pencarian buku. Yang akan dibahas adalah konsep pengembangan *mobile agent*, cara pembuatan, dan penerapannya dalam suatu aplikasi. Sistem pencarian buku adalah suatu sistem yang memungkinkan pengguna untuk mencari judul buku pada beberapa basisdata *server* sekaligus. Jadi pengguna tidak harus masuk ke dalam situs toko buku satu per satu tetapi dengan menggunakan sistem ini pengguna cukup memasukan kriteria dari buku yang akan dicari dan nantinya *mobile agent* akan mencari di setiap basisdata *server*. Gambar 3 adalah gambar topologi sistem yang akan dibuat.



Gambar 3 Topologi sistem pencarian buku

Client akan mengirimkan *mobile agent* sesuai dengan jumlah basisdata *server* yang aktif. Pada studi kasus kali ini menggunakan dua buah basisdata *server* yaitu dengan nama basisdata *server* toko A dan basisdata *server* toko B. *Mobile agent* tersebut akan melakukan pencarian pada masing-masing basisdata *server* dan kembali ke komputer *client* lalu memperbaharui basisdata dan informasi yang didapat ditampilkan di *web browser client*. Sedangkan arsitektur dari sistem monitoring komputer bisa dilihat pada Gambar 4.



Gambar 4 Arsitektur sistem pencarian buku

Mobile agent yang digunakan berjumlah tiga *agent* yaitu *AgentOne*, *AgentTwo* dan *AgentThree*. *AgentOne* akan mengambil *keyword* buku dari pengguna yang dimasukan lewat *browser*. Setelah itu *AgentOne* akan membuat *AgentTwo* dan *AgentTwo* akan berpindah dari sisi *client* ke sisi basisdata *server*. Setelah sampai di basisdata *server*, *AgentTwo* akan melakukan proses *query* dengan menggunakan *keyword* untuk mendapatkan buku yang diinginkan pada basisdata *server*. Setelah itu *AgentTwo* akan membuat *AgentThree* dan memasukkan hasil pencarian ke *AgentThree* kemudian mengirimkan *AgentThree* kembali ke sisi *client*. Setelah sampai di sisi

client, *AgentThree* akan memperbaharui basisdata dengan hasil yang dibawanya.

4. Hasil Pengujian dan Analisis

4.1 Spesifikasi Penujian

1. Spesifikasi Komputer

Pada pengujian kali menggunakan tiga buah laptop dimana masing-masing laptop mempunyai tugas-tugas sendiri. Laptop pertama akan digunakan sebagai klien dengan spesifikasi sebagai berikut :

- a. Prosesor Intel Atom 1,6 GHz
- b. Harddisk 320 GB
- c. RAM 2 GB

Laptop kedua akan digunakan sebagai basisdata *server* toko buku A dengan spesifikasi sebagai berikut :

- a. Prosesor Intel Core 2 Duo 2 GHz
- b. Harddisk 160 GB
- c. RAM 2 GB

Laptop ketiga akan digunakan sebagai basisdata *server* toko buku B dengan spesifikasi sebagai berikut :

- a. Prosesor Intel Core i5 2 GHz
- b. Harddisk 500 GB
- c. RAM 2 GB

2. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak untuk pengujian sistem ini diperlukan karena sistem ini merupakan sistem yang dijalankan dengan metode khusus yaitu *Mobile Agent*. Spesifikasi perangkat lunak yang digunakan adalah :

- a. LAMP (Linux Apache MySQL PHP) dimana Linux yang digunakan adalah Ubuntu Desktop Edition versi 9.10 dan 11.04, Apache versi 2.2.12, MySQL versi 5.1.37 dan PHP versi 4.4.6.
- b. *Framework* Aglets 2.0.2

4.2 Pengujian Sistem Pencarian Buku

Pengujian pencarian buku bertujuan apakah fungsi pencarian buku bisa dilakukan dengan baik karena pencarian buku merupakan inti dari penelitian menggunakan metode *Mobile Agent*. Pada pengujian ini ada dua basisdata *server* dengan nama toko adalah toko buku A dan toko buku B . Proses awal yaitu dengan memasukan *keyword* dari buku yang akan dicari di menu *search* pada halaman *search.php* seperti pada Gambar 5

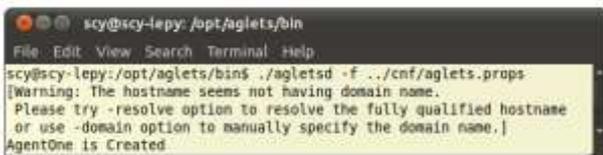


Gambar 5 Tampilan masukan keyword

Langkah selanjutnya yaitu menjalankan Tahiti server. Setelah masuk ke Tahiti server, buat *AgentOne* seperti pada Gambar 6. Tampilan *AgentOne* telah terbuat juga bisa dilihat di console Tahiti server seperti pada Gambar 7 Setelah *AgentOne* terbuat, *AgentOne* akan membuat *AgentTwo* seperti pada Gambar 8. *AgentTwo* kemudian berpindah ke tiap komputer basisdata server.



Gambar 6 Tampilan create *AgentOne*

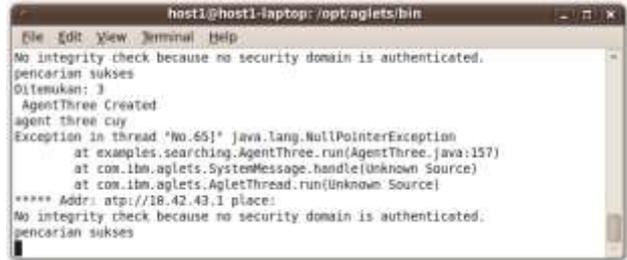


Gambar 7 Tampilan console buat *AgentOne*

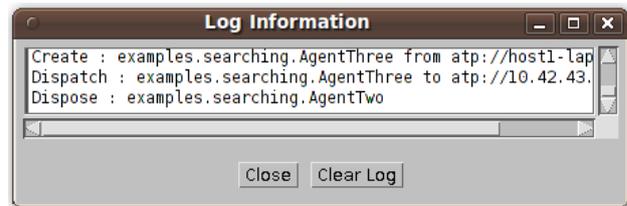


Gambar 8 Tampilan *AgentOne* membuat *AgentTwo*

Setelah sampai di basisdata server, *AgentTwo* akan melakukan pencarian judul buku sesuai dengan keyword yang dibawa. *AgentTwo* akan memberikan pemberitahuan jika menemukan buku yang dicari di console Tahiti server seperti pada Gambar 9. Kemudian *AgentTwo* akan membuat *AgentThree*. Proses tersebut bisa dilihat pada log information dari Tahiti Server seperti pada Gambar 10. Lalu *AgentThree* akan berpindah ke komputer klien yang mengirimnya.

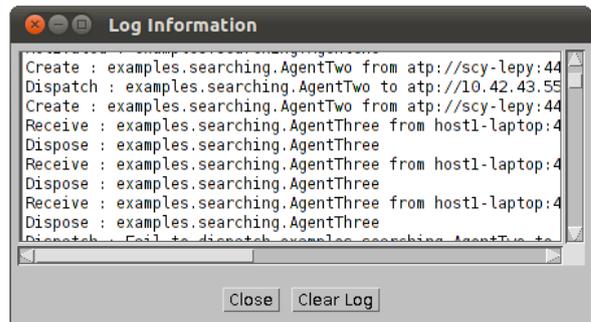


Gambar 9 Tampilan pemberitahuan menemukan buku



Gambar 10 Tampilan *AgentTwo* membuat *AgentThree*

Setelah *AgentThree* kembali ke komputer klien, *AgentThree* akan memperbaharui basisdata dengan hasil yang dibawanya dan menghapus dirinya sendiri. Proses tersebut bisa dilihat di log information dari Tahiti server seperti pada Gambar 4.11. Untuk melihat hasil pencarian buku yang dilakukan oleh *Mobile Agent*, pengguna harus membuka browser dengan alamat <http://localhost/searching>. Hasil dari pencarian ada tiga yaitu ada buku yang ditemukan pada dua toko, ada buku yang ditemukan di salah satu toko sedangkan toko lainnya tidak ada dan tidak ditemukan buku pada basisdata kedua toko tersebut.



Gambar 11 Tampilan *AgentThree* dispose

1. Buku Ditemukan di Kedua Basisdata Toko

Jika buku ditemukan di basisdata tokobuku A dan B maka akan tampil daftar judul buku yang ditemukan dengan keterangan dari nama dua toko tersebut seperti pada Gambar 12



Gambar 4.12 Tampilan buku ditemukan

2. Buku Ditemukan di Salah Satu Basisdata Toko

Jika *Mobile Agent* hanya menemukan buku di salah satu toko maka hanya akan tampil daftar judul buku dari salah satu toko dan simbol “-” di salah satu baris yang menandakan bahwa pencarian *Mobile Agent* pada salah satu basisdata toko tidak menemukan judul yang dicari. Hasilnya bisa dilihat seperti pada Gambar 13.



Gambar 13 Tampilan buku ditemukan di salah satu toko

3. Buku Tidak Ditemukan di Kedua Basisdata Toko

Jika *Mobile Agent* tidak menemukan judul buku yang dicari pada semua basisdata toko maka tampilan dari hasil tersebut yaitu simbol “-”. Tampilannya seperti pada Gambar 14.



Gambar 14 Tampilan buku tidak ditemukan

4.3 Hasil Pengujian Secara Keseluruhan Sistem

Berikut ini adalah hasil pengujian terhadap sistem secara keseluruhan

Tabel 1 Hasil pengujian keseluruhan Sistem

No	Uji	Hasil Pengujian Sukses
1.	Pencarian Buku dengan kondisi menemukan buku di semua basisdata server	√
2.	Pencarian Buku dengan kondisi menemukan buku di salah satu basisdata server	√
3.	Pencarian Buku dengan kondisi buku tidak ditemukan di semua basisdata server	√

5. Penutup

5.1 Kesimpulan

Dari hasil analisa dan pembahasan dapat disimpulkan bahwa :

1. Aplikasi sistem pencarian buku ini dapat diimplementasikan dengan menggunakan konsep *Mobile Agent* dengan menggunakan *framework* aglets 2.0.2 dan menggunakan Linux Ubuntu sebagai sistem operasinya.
2. Kriteria pencarian buku dibagi menjadi tiga bagian yaitu judul, pengarang, dan penerbit. Dalam hal ini, untuk melakukan pencarian bisa mengkombinasikan dari ketiga kriteria tersebut.
3. Pada sistem pencarian buku ini diberikan tiga buah hasil yaitu ditemukan buku pada semua basisdata toko, ditemukan buku di salah satu basisdata toko dan tidak ditemukan buku pada semua basisdata toko.
4. Sistem ini menggunakan tiga buah agent dalam proses pencarian buku yaitu *AgentOne*, *AgentTwo* dan *AgentThree*. Masing-masing agent mempunyai tugasnya sendiri-sendiri. *AgentOne* berfungsi untuk menyimpan *keyword* buku yang akan dicari dan membuat *AgentTwo*. *AgentTwo* berfungsi untuk mencari buku di masing-masing basisdata toko sedangkan *AgentThree* berfungsi memperbaharui basisdata hasil pencarian di komputer klien. Ketiga

agent tersebut membentuk kolaborasi program yang baik sehingga sistem pencarian buku ini dapat berjalan sesuai yang diharapkan

5. Sistem pencarian buku menggunakan *Mobile Agent* ini hanya dapat berjalan pada komputer yang sudah terpasang Aglets dan Java.

5.2 Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

1. Perlu dilakukan penelitian lebih lanjut tentang keamanan data yang dibawa oleh agen, karena bagaimanapun juga jaringan komputer adalah tempat yang sangat rentan dalam hal keamanan dan kerahasiaan pertukaran data.
2. Perlu dilakukan penelitian lebih lanjut untuk mengetahui bagaimana agen dapat dibuat tidak bergantung pada Tahiti Server.
3. Menambahkan tidak hanya proses mencari buku saja tetapi ada proses lain karena sistem pada penelitian ini merupakan sebuah inti dari sistem yang lebih besar dan bisa dikembangkan lebih lanjut.
4. Sistem bisa diterapkan lintas sistem operasi. Misalnya antara Windows dengan Linux.

Daftar Pustaka

- [1]. Aglets Development Group, "The Aglets 2.0.2 User's Manual", 2009.
- [2]. Aldilla, R. "Perancangan dan Pembuatan Perangkat Lunak Pencarian dan Pemesanan Tiket Pesawat Berbasis Teknologi Mobile Agent", Institut Teknologi Sepuluh November, Surabaya, 2007.
- [3]. Bond, A.H. and L. Gasser, "Readings in Distributed Artificial Intelligence", *Morgan Kaufmann Publishers*, 1988.
- [4]. Chaib-draa, B., R.B. Moulin, Mandiau, and P. Millot, "Trends in Distributed Artificial Intelligence", *Artificial Intelligence Review*, 6, 35-66, 1992.
- [5]. Dasar-dasar PHP, http://www.siaksoft.net/index.php?option=com_content&ask=view&id=812&Itemid=47, Juni 2011.
- [6]. Gasser, L. and M. Huhns, "Distributed Artificial Intelligence", Vol. 2, *Morgan Kaufmann Publishers*, 1989.
- [7]. Guralnik, D.B. "Webster's New World dictionary", 1983.
- [8]. Hewitt, C. "Viewing Control Structures as Patterns of Passing Messages", *Artificial Intelligence*, 8(3), pp.323-364, 1977.
- [9]. Kulin, A. "A Distributed Security Management System Based on Mobile Agents", Universitas Teknik Wina, 2001.
- [10]. Lange, D.B. and M. Oshima, "Programming And Deploying Java *Mobile Agents* With Aglets", Addison-Wesley, 1998.
- [11]. Lange, D.B. "Mobile Object and Mobile Agent: The Future of Distributed Computing?", General Magic Inc, California.
- [12]. Oshima, M., K. Guenter, K. Ono, Aglets Specification 1.1 Draft., IBM Corp, www.trl.ibm.co.m/aglets/spec11.html
- [13]. Nwana, H. and D. Ndumu, "An Introduction to Agent Technology", *BT Technology Journal*, 14(4), 1996.
- [14]. Somantri, M. "Mobile Computing in Fact", 4th ICTS, Teknik Informatika ITS, 2008.
- [15]. Tim Litbang LPKBM MADCOMS, "Mendesain Website dengan Photosop, Frontpage dan Pemrograman PHP-MySQL", Hal 198, 2008
- [16]. Wooldridge, M.J. and N.R. Jennings, "Agent Theories, Architectures, and Languages: A Survey", *Proceedings of the Workshop on Agent Theories, Architectures, and Languages (ECAI-94)*, 1994.
- [17]. Wooldridge, M.J. and N.R. Jennings, "Intelligent agents: Theory and Practice", *Knowledge Engineering Review*, 10(2), 1995.
- [18]. Wooldridge, M.J., J.P. Mueller, M. Tambe, "Intelligent Agents II", *Lecture Notes in Artificial Intelligence 1037*, Springer-Verlag, 1996.