REKAYASA PERANGKAT LUNAK BERORIENTASI AGEN SOLUSI UNTUK KEBERLANGSUNGAN HIDUP PERANGKAT LUNAK DAN MENJAMIN KUALITAS PERANGKAT LUNAK

A r a d e a[1], *Asep Andang*[2]

[1] *Jurusan Teknik Informatika Fakultas Teknik*
[2] *Jurusan Teknik Elektro Fakultas Teknik*
Universitas Siliwangi Tasikmalaya

I. PENDAHULUAN

Perangkat lunak yang didesain untuk melakukan pekerjaan pada sebuah lingkungan tertentu, memiliki kecenderungan pada suatu saat, tidak lagi sesuai dengan perilaku lingkungan disekitarnya. Hal ini disebabkan karena terjadi perubahan lingkungan. Pada kondisi tersebut, kehidupan dari perangkat lunak perlahan-lahan akan berakhir atau mati, yang kemudian akan digantikan oleh perangkat lunak yang baru. Maka perubahan lingkungan dapat mengancam keberlangsungan hidup sebuah perangkat lunak. Perubahan lingkungan yang dimaksud, seperti pertambahan proses pada prosedur kerja, contohnya jika diperlukan penambahan *approve* transaksi tertentu dari manager, yang dulunya hanya dari supervisor. Dalam hal ini, perangkat lunak harus menambahkan proses tersebut. Akan ditemukan kesulitan jika programmer harus mengabiskan waktu, untuk mempelajari perangkat lunak tersebut, karena jika ketika menambah suatu proses mungkin akan mengganggu proses lainnya. Kemudian perubahan lingkungan dapat juga berupa pengetahuan yang ada di lingkungan yang terus berkembang.

Rekayasa perangkat lunak berorientasi agen (*Agent Oriented Software Enginering*, AOSE), merupakan paradigma baru bagi para peneliti bidang rekayasa perangkat lunak, untuk melakukan, menghasilkan analisis dan sintesis setiap produk perangkat lunak sistem. Metodologi AOSE merupakan suatu pendekatan dalam upaya mengembangkan dan membangun perangkat lunak dengan menggunakan abstraksi berorientasi-agen secara alamiah dengan memodelkan sebuah sistem kompleks seperti dunia nyata. Suatu sistem kompleks dapat dipandang terdiri dari subsistem-susbsistem atau agen, interaksi agen-agen, atau berprilaku seperti agen-agen didalam suatu organisasi.

Metodologi AOSE menjanjikan kemampuan untuk membangun sebuah sistem secara sangat fleksibel. Dimana kompleksitas dan prilaku persoalan dibangun degan pengkombinasian sangat canggih, modular dari komponen-komponen *intelligent*. Sifat-sifat cerdas dari sebuah sistem (perangkat lunak agen, atau cukup '*agent*' saja) dapat berupa memiliki kemampuan belajar, merencanakan, berkomunikasi dan bernegosiasi.

II. LANDASAN TEORI

2.1. Agen dan Multiagen

Di dalam kamus *Webster's New World Dictionary*, agent didefinisikan sebagai: *A person or thing that acts or is*

capable of acting or is empowered to act, for another.

Disini ada dua point yang bisa diambil :

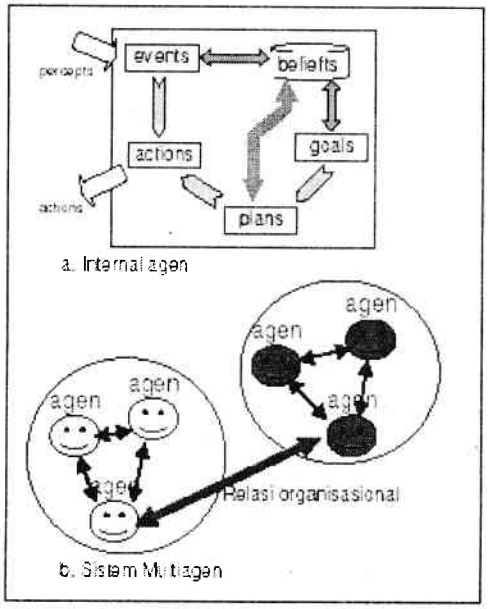
- Agen mempunyai kemampuan untuk melakukan suatu tugas/pekerjaan.
- Agen melakukan suatu tugas/pekerjaan dalam kapasitas untuk sesuatu, atau untuk orang lain.

Ditarik dari point-point diatas dapat didefinisikan *software agent* sebagai: Suatu entitas software komputer yang memungkinkan *user* (pengguna) untuk mendelegasikan tugas kepadanya secara mandiri (*autonomously*).

Kemudian beberapa peneliti lain menambahkan satu point lagi, yaitu bahwa *agent* harus bisa berjalan dalam kerangka lingkungan jaringan (*network environment*). Definisi *agent* dari para peneliti lain pada hakekatnya adalah senada, meskipun ada yang menambahkan atribut dan karakteristik *agent* kedalam definisinya.

Dalam perkembangan aplikasi dan penelitian tentang *agent*, bagaimanapun juga dalam suatu komunitas sebuah sistem tidak dapat dihindari akan dibutuhkannya lebih dari satu *agent*, seiring dengan semakin kompleksnya tugas yang dikerjakan oleh sistem tersebut. Dilain pihak, dengan populernya paradigma *agent* terjadi pembengkakan populasi *agent*, karena setiap vendor ataupun pembuat software berkeinginan untuk memakai paradigma *agent* dalam sistem mereka. Disinilah kemudian dimulai usaha untuk standarisasi *agent*, dengan tujuan untuk mendukung pertumbuhan *agent* dan *multi agent system* supaya terarah dan lebih terbuka. Usaha untuk melakukan standarisasi terhadap software agent, baik secara fisik maupun secara teori ini terbentuknya beberapa organisasi yang melakukan usaha standarisasi, antara lain Organsiasi yang terbesar adalah *Foundation for Intelligent Physical Agent* (FIPA), kemudian *Object Management Group* (OMG), *US Defense Advanced Research Projects Agency* (DARPA), dan *AgentLink*.

Hal yang penting lain bahwa suatu saat diharapkan bahwa meskipun berbeda vendor dan pembuat, antara satu agent dan agent lain bisa tetap berkomunikasi dan berkoordinasi berhubungan dengan suatu pekerjaan. Paradigma pengembangan sistem dimana dalam suatu komunitas sistem terdapat beberapa *agent*, yang saling berinteraksi, bernegosiasi dan berkoordinasi satu sama lain dalam menjalankan pekerjaan, disebut dengan *multi agent system* (MAS).

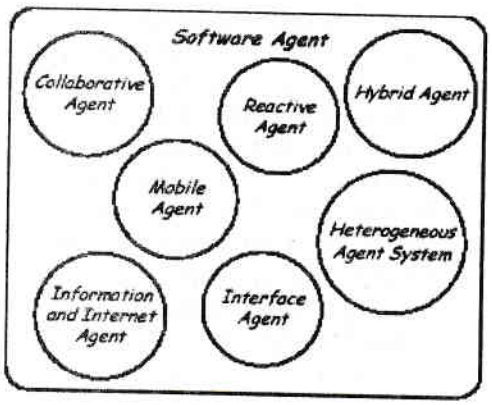


Gambar 1. Model Perangkat Lunak Berbasis Agen

Pada gambar 1, diperlihat skema sifat-sifat, karakteristik model sebuah agen dan sistem multiagen secara organisasi. Sebuah agen didasarkan pada sekumpulan pengetahuan terpercaya (*beliefs, siqnificant knowledge*). Dengan pengetahuan, agen mempunyai kemampuan beragumentasi (*actions*). Pengetahuan agen direpresentasikan secara lebih umum dari pada basis-aturan (*rules*), dalam bentuk atribut-atribut, status, dan kepemilikan. Sebuah agen memiliki sifat-sifat autonomous dalam pengertian bahwa agen bekerja atau melayani diri sendiri dan memiliki kepentingan tanpa interpensi langsung secara ekternal. Menurut hasil penalarannya agen dapat memutuskan sendiri dan tindakan kapan akan dilaksanakan (*plans, and actions*). Jika agen menyimpulkan sesuatu, dimungkinkan untuk memberitahukan kepada agen-agen lain. Prilaku autonomous ini diturunkan dari mesin aturan yang secara proaktif terus mengevalasi aturan. Sebuah agen memiliki tujuan jelas (*well-define goal*) yang disajikan oleh pengetahuan dan prilaku yang dimilikinya (yaitu kemampuan memberikan alasan berdasarkan pengetahuan). Sebuah agen mencoba memberikan alas an sederhana (*actions*), memberikan pengetahuannya, untuk memenuhi tujuannya.

2.2. Klasifikasi Agen

Agent menurut karakteristik yang dimilikinya bisa diklasifikasikan menjadi delapan.



Gambar 2. Klasifikasi *Agent* Menurut Karakteristik Yang Dimiliki

1. Collaborative Agent: *Agent* yang memiliki kemampuan melakukan kolaborasi dan koordinasi dengan agent lain dalam kerangka *Multi Agent System* (MAS).
2. Interface Agent: *Agent* yang memiliki kemampuan untuk berkolaborasi dengan user, melakukan fungsi *monitoring* dan *learning* untuk memenuhi kebutuhan user.
3. Mobile Agent: Agent yang memiliki kemampuan untuk bergerak dari suatu tempat ke tempat lain, dan secara mandiri melakukan tugas ditempat barunya tersebut, dalam lingkungan jaringan komputer.
4. Information dan Internet Agent: *Agent* yang memiliki kemampuan untuk menjelajah internet untuk melakukan pencarian, pemfilteran, dan penyajian informasi untuk user, secara mandiri. Atau dengan kata lain, memanage informasi yang ada di dalam jaringan Internet.
5. Reactive Agent: *Agent* yang memiliki kemampuan untuk bisa cepat beradaptasi dengan lingkungan baru dimana dia berada.

6. Hybrid Agent: *Agent* yang memiliki katakteristik yang merupakan gabungan dari karakteristik yang sudah disebutkan sebelumnya adalah masuk ke dalam *hybrid agent*.
7. Heterogeneous Agent System: Dalam lingkungan *Multi Agent System* (MAS), apabila terdapat dua atau lebih *hybrid agent* yang memiliki perbedaan kemampuan dan karakteristik, maka sistem MAS tersebut kita sebut dengan *heterogeneous agent system*.

2.3. *Intelligent Agents* dan *Expert System*

Expert system dikenal sebagai sebuah program komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Kemampuan menyelesaikan masalah pada *expert system* sangat bergantung pada pemindahan pengetahuan dari *human expert* ke *expert system*. Pemindahan kepakaran adalah mentransfer kepakaran yang dimiliki seorang pakar kedalam komputer. Aktivitas yang dilakukan untuk memindahkan kepakaran, meliputi :

a. Knowledge Acuisition
b. Knowledge Representation
c. Knowledge Inferecing
d. Knowledge Transfering

Knowledge Acuisition

Knowledge Acuisition meliputi proses pengumpulan, pemindahan, dan perubahan dari pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi (buku dan lain-lain) ke program komputer yang bertujuan untuk memperbaiki dan atau mengembangkan basis pengetahuan (*knowledge base*)

Knowledge Representation

Manusia memperoleh pengetahuan melalui pengalaman dengan melihat, mendengar, merasakan. Maka dapat dikatakan, apa yang kita kita lihat dan kita dengar membawa pengetahuan yang baru. Sehingga pengetahuan adalah fakta atau kondisi tentang sesuatu hal. Kemudian bagaimana kita menyimpan pengetahuan ini dalam sebuah komputer merupakan masalah yang ditangani oleh *knowledge representation*. Terdapat cara untuk merepresentasikan pengetahuan, antara lain:

a. *Procedural*, prosedural code bukan hanya mengkodekan fakta, tetapi juga mendefinisikan urutan operasi dan manipulasi fakta. Pada procedural code, pengetahuan dan manipulasinya tidak dimungkinkan saling berhubungan. Kelemahan ini diperbaiki oleh pendekatan yang disebut declaration.
b. *Declaration*, pada pendekatan ini, fakta, aturan, keterhubungan direpresentasikan dalam bentuk pengetahuan yang utuh.

Knowledge Inferecing

Proses inferensi adalah proses yang digunakan dalam sistem pakar untuk menghasilkan informasi baru dari informasi yang telah diketahui. Dalam sistem pakar dilakukan oleh mesin inferensi (*Inference Engine*). Mesin ini merupakan modul yang berisi program tentang bagaimana mengendalikan proses reasoning. Proses reasoning yang dimaksud adalah proses bekerja dengan pengetahuan dan fakta untuk mengambil suatu keputusan. Adapun metode reasoning meliputi:

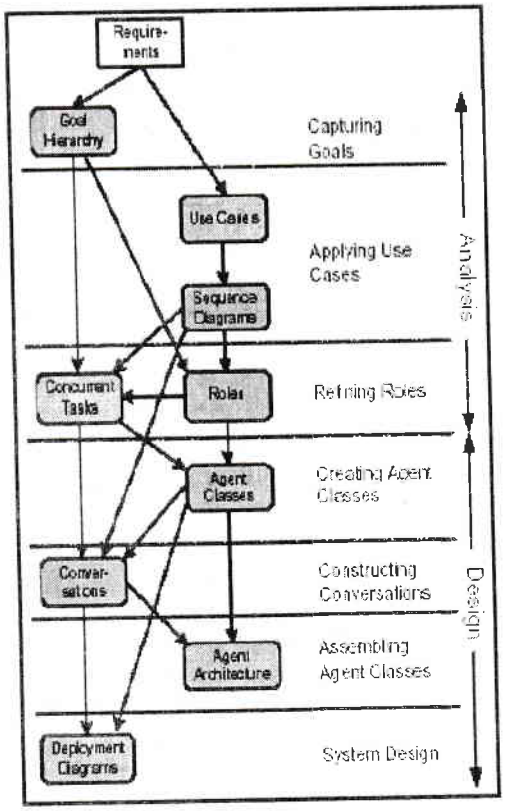
a. *Deductive Reasoning*, mendeduksi informasi baru dari hubungan logika pada informasi yang telah diketahui.
b. *Induktive Reasoning*, mengambil kesimpulan umum dari sejumlah fakta khusu tertentu.

c. *Abductive Reasoning*, bahwa kesimpulan mungkin bisa mengikuti informasi yang tersedia, tetapi juga bisa salah.
d. *Analogical Reasoning*, menggambarkan analogi antara 2 objek/situasi, kemudian melihat persamaan dan perbedaan untuk memandu proses reasoning.
e. *Common Sense Reasoning*, dengan pengalaman manusia belajar memecahkan masalahnya secara cepat, dalam expert system dinamakan *heuristic search* atau *best first search.* Inferensi dengan rules, merupakan implementasi dari modus ponen, yang direfleksikan dalam mekanisme pencarian (*search*). Ada dua metode *inferencing* dengan *rules*, yaitu meliputi: *Backward Chaining*, pendekatan goal driven, dimulai dari ekspektasi apa yang diinginkan terjadi (hipotesa), kemudian mengecek pada sebab-sebab yang mendukung atau menolak dari ekspektasi tersebut. *Forward Chaining*, dengan pendekatan yang berbeda dari backward chaining, forward chaining melakukan pencarian dari suatu masalah kepada solusinya.

III. METODE PENELITIAN

Banyak pendekatan telah diusulkan oleh para ahli untuk membangun perangkat lunak berbasis agen. Secara umum, meskipun cara penanganan berbeda namun metodologi-metodologi tersebut saling melengkapi. Diantaranya adalah Metodologi MaSE, pembangunan merupakan konsep interaksi multiagen.

Metodologi rekayasa sistem multiagen (*Multiagent System Engineering*, MaSE), dimulai dengan langkah spesifikasi sistem, dan selanjutnya memproduksi sekumpulan dokumen rancangan formal dalam bentuk-bentuk grafis. Perhatian utama dari metodologi MaSE adalah untuk membantu para pengembang dalam merancang dan membangun perangkat lunak berbasis sistem multiagen, melalui siklus hidup perangkat lunak dari proses spesifikasi hingga implementasi sistem multiagen. Sebuah sistem yang dirancang dengan metodologi MaSE dapat diimplementasikan dalam berbagai cara dari rancangan yang sama. Setiap rancangan obyek dapat dilacak kembali dari/ke aktivitas sebelumnya dan dari/ke aktivitas berikutnya pada setiap tahapan.



Gambar 3. Tahapan Metodologi MaSE

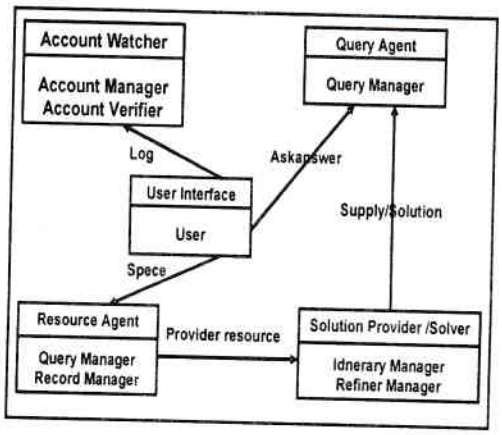
Secara singkat tahapan dan model dari metodologi MaSE diperlihatkan pada gambar 3. Tahapan proses-proses dari metodologi MaSE terdiri dari tujuh langkah aktivitas, yang dikelompokkan dalam dua phase dan memproduksi sepuluh model dokumen formal. Phase Analisis terdiri dari langkahlangkah seperti 1). *Capturing Goal*, 2) *Applying Use Cases*, 3) *Refining Roles*. Phase Desain terdiri dari langkah-langkah 4) *Create Agent Classes*, 5) *Contruction Conversations*, 6) *Assembling Agent Classes*, dan 7) *System Design*.

3.1. Phase Analisis MaSE

a. *Capturing Goals*, langkah yang membantu para analis untuk mengidentifikasi tujuan-tujuan, struktur dan menyajikannya sebagai hirarki/berjenjang tujuan sistem.

b. *Applying Use Cases*, meliputi penyerapan scenario utama dari konteks awal sistem. *Use Cases* juga digunakan untuk membangun diagram terurut (mirip seperti pada UML).

c. *Refining Roles*, merupakan langkah terakhir dari tahapan analisis, yaitu untuk pembentukan model peran dan model tugas konkuren. Model peran menggambarkan peranan dari agen dalam sistem. Di samping itu, peran-peran yangdibentuk juga harus menunjukan bagaimana tujuan-tujuan sistem akan dicapai dan jalur komunikasi diantara peran-peran tersebut. Model tugas yang konkuren digambarkan dalam bentuk diagram status mesin.



Gambar 4. Contoh diagram class agen

3.2. Phase Desain MaSE

a. *Creating Agent Classes*, hasil dari langkah adalah sebuah diagram class agen yang mengambarkan secara menyeluruh sistem multiagen. Diagram class agen memper-lihatkan agen-agen dan peranan yang dimainkannya. Hubungan (*link*) antar agen-agen diperlihatkan melalui notasi percakap-an, misalnya seperti yang ditunjukkan pada gambar 5.

b. *Construction Conversation*, merupakan diagram detail mekanisme konversasi dari agen-agen yang digambarkan dalam diagram komunikasi classes, yang berbentuk mesin status berhingga (*finitestate machine*)

c. *Assembling Agent Classes*, pada langkah ini didefinisikan aristektur agen dan komponen-komponen penyusunnya.

D. *System Design*, meliputi pembuatan digram *deployment*, yang menspesifikan penempatan agen-agen dalam sebuah sistem.

IV. HASIL DAN PEMBAHASAN

Semua yang ada di muka bumi ini tidak ada yang abadi, tetapi bukan berarti manusia dilarang berusaha untuk mempertahankan hidup. Karena sesungguhnya manusia telah diberikan akal oleh Sang Maha Pencipta, untuk berusaha mencari reski dan berpikir demi mempertahankan hidupnya. Manusia mempertahankan hidupnya untuk dapat memberikan manfaat lebih banyak ke orang-orang disekitarnya dan lingkungannya, dengan terus memperbaiki diri untuk menjadi yang lebih baik. Tetapi kematian pasti akan datang, tetapi tidak untuk ditunggu, tapi dihadapi. Hal di atas, dapat dijadikan inspirasi dalam mengembangkan perangkat lunak. Tidak ada perangkat lunak yang hidup selamanya, satu saat pasti akan mati. Tetapi bukan berarti perangkat lunak itu diam dan menunggu mati. Perangkat lunak itu harus bertahan dan berusaha untuk menjaga kerberlangsungan hidupnya, dengan cara menambah pengetahuannya, memberikan aksi terhadap informasi dan perubahan lingkungannya, sehingga dapat terus memberikan manfaat bagi pengguna dan lingkungannya.

Perangkat lunak yang dibangun dengan multi-agent system, akan dapat mengelola pengetahuan dan pengalamannya, dalam menyelesaikan tugas yangdidelegasikan kepadanya. Karena muti-agent system terdiri dari agent-agent yang handal dan cerdas (*Intelligent Agent*). Agent-agent tersebut dapat menerima dan mengenal informasi yang diterimanya, serta melakukan aksi terhadap sesuatu yang diterimanya dari pengguna, lingkungan, maupun dari agent lainnya. Sehingga dapat dikatakan perangkat lunak yang dibangun dengan *multi-agent system*, akan dapat bertahan dari kematian, dan dapat menjaga keberlangsungan hidupnya, dengan selalu meningkatkan pemanfaatannya kepada lingkungan dan pengguna. Perangkat lunak yang tidak dapat ditambah pengatahuannya, sehingga tidak dapat memberikan manfaat yang tinggi kepada lingkungannya, akan segera mati dan digantikan oleh perangkat lunak yang lain. Tetapi kejadian tersebut tidak akan terjadi kepada perangkat lunak yang dibangun dengan agent-agent yang cerdas (*Intelligent Agents*).

Dalam dunia kesehatan salah satunya dikenal e-medicine. E-medicine adalah penerapan IT (*Information Technology*) dalam dunia kesehatan. E-medicine meliputi *telemedicine*, *e-healthcare, e-diagnosis, econsultation, e-clinic*, dan lain-lain. Dalam hal ini, akan dibahas pembangunan telemedicine dengan pendekatan *Multi-Agent System.* Pendekatan *Multi-Agent System* dalam pembangunan telemedicine, dilakukan dengan langkah-langkah sebagai berikut:

1. *Requirement Analysis*
2. Desain
   a. Desain *Goals* and *Functional Specification*
   b. Desain Agent
   c. Desain *Multi Agent Society*

Requirement Analysis

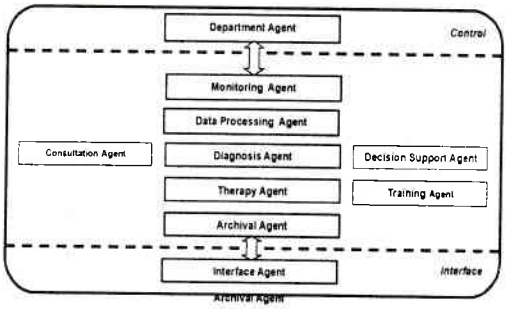
Sistem telemedicine harus dapat memberi pemantauan dan pelayanan setiap saat kepada pasien. Adapun analisis kebutuhan, sebagai berikut:

a. Menangani kunjungan ke pasien dan pelaksanaan terapi personal.
b. Pamantauan terhadap kondisi pasien setiap waktu tertentu, serta pemantauan datanya.

c. Mendiagnosis pasien, berdasarkan hasil pemantauan kesehatan dan datanya.
d. Melakukan penyuluhan untuk pasien, agar dapat memahami hal-hal penting yang harus diperhatikan, berkaitan dengan penyakitnya.
e. Membangun sistem database yang baik, untuk mengelola data-data pasien.
f. Mengelola interaksi telemedicine dengan sistem lainnya

Desain Goals and Functional Specification

Pendekatan *Multi-agent system* dalam pembangunan Telemedicine, terdiri dari tiga kelompok tujuan, yaitu *interface group, implementation group dan control group*. Dalam implementation group, terdapat beberapa *agent* yang memiliki tanggung jawab yang berbeda. Dibawah ini, ditampilkan arsitektur dari sistem *multi agent*.



Gambar 5. Arsitektur Multi Agent System dari Telemedicine

Pada sistem *multi agent* ini, *control group* memiliki tugas sebagai mediator, jika terjadi konflik antar agent. Sedangkan interface group, menjaga hubungan *agent* dengan pasien dan antar bagian pada sistem *telemedicine*. Implementation group mengimplementasikan pemantauan, diagnosis, terapi, konseling, dan mengupayakan agar tujuan dapat tercapai.

Desain *Agent*

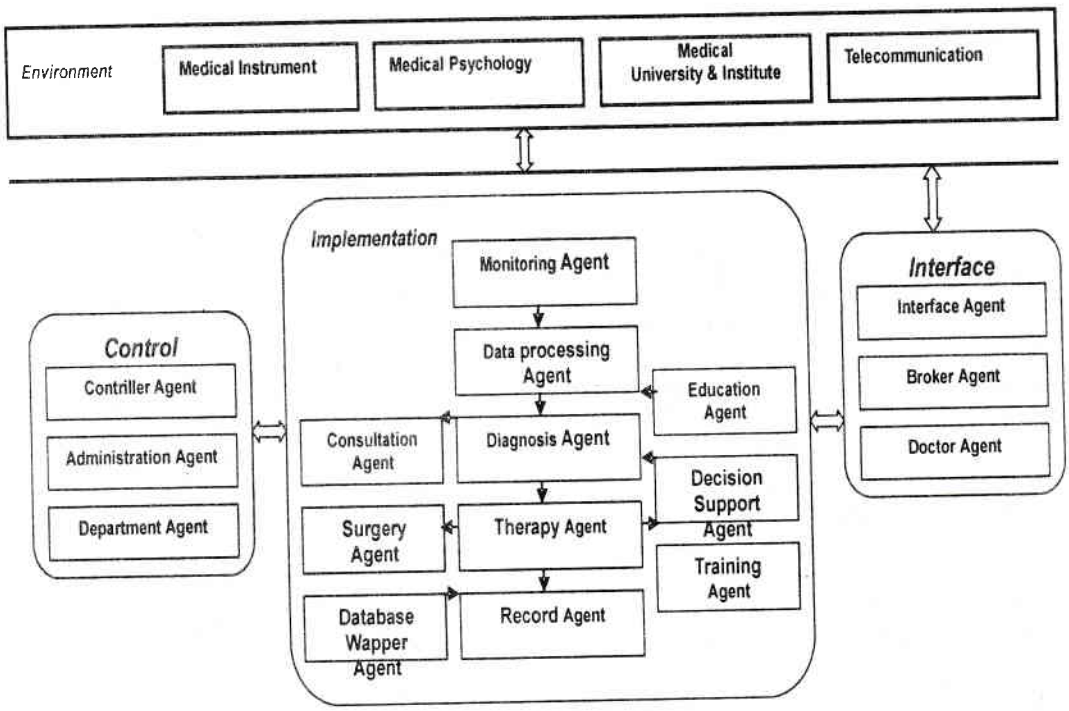
Berdasarkan kebutuhan yang disebutkan di atas, maka dapat didesain beberapa *agent*, sebagai berikut: *monitoring agent, data processing agent, consultation agent, traning agent, archical agent, departement agent dan interface agent*. Adapun tanggung jawab *agent* tersebut, sebagai berikut:

a. *Monitoring agent*, bertanggung jawab memantau kondisi kesehatan pasien setiap saat tertentu, serta mengirimkan data hasil pemantauan pasien ke data processing agent.
b. *Data processing agent*, membuat statisitk dan integrasi data hasil pemantauan.
c. *Diagnosis agent*, menganalisis situasi dan kondisi pasien, serta dapat menentukan hal-hal yang penting berkaitan penyakit tertentu.
d. *Therapy agent*, menentukan metode terapi yang sesuai dengan kondisi pasien.
e. *Consultation agent*, menyelenggarakan konseling untuk pasien, berkaitan dengan hasil diagnosis dari diagnosis agent.
f. *Decision support agent*, melaksanakan dukungan terhadap pengambil keputusan, serta bekerja sama dengan *diagnosis agent*.
g. *Traning agent*, melatih pasien untuk memahami hal-hal penting mengenai penyakit yang dideritanya, serta bagaimana menjaga kondisi dapat kembali baik.
h. *Archival agent*, melakukan menambahan atau perubahan terhadap data pasien dan metode terapi yang sedang dijalankan. Kemudian menintegrasikan dengan database individu pasien.

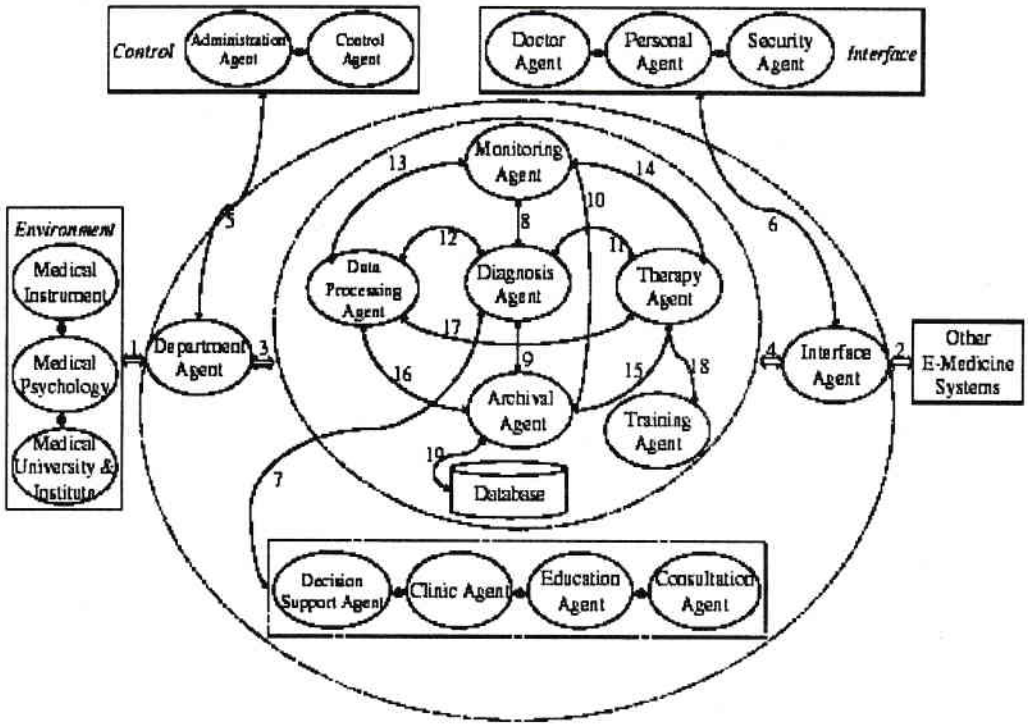
i. *Departement agent*, mengimplementasikan kontrol pada jalannya sistem telemedicine.
J. *Interface agent*, membantu layanan pencarian data dan informasi.

Desain *Multi-Agent System Society*

Desain *multi agent society* fokus pada pendirian arsitektur sistem *multi agent* dan interaksi antar *agent*. Model interaksi dalam sistem *multi agent* dibagi menjadi model eksternal dan model internal. Model interaksi internal dari system *telemedicine* bukan hanya antara agent dengan satu departemen, tetapi juga dengan beberapa departemen yang berbeda-beda. Model interaksi eksternal misalnya *medical instrument*, dukungan bidang keilmuan psikologi, peran



Gambar 6. Multi-Agent Society



Gambar 7. Interaksi pada Multi-Agent System

Panah 1 dan 2, menggambarkan bahwa telemedicine lingkungan di luarnya dan system telemedicine lainnya yang berkaitan. Panah 3 dan 4 menggambarkan *implementation group* berinterakasi dengan *control group* dan *interface group*, dalam sistem *telemedicine.* Panah 5, merepresentasikan interaksi antara *department agent* dan *control group* dalam *telemedicine*, seperti interaksi antara departemen administrasi dan control group. Panah 6, merepresentasikan interaksi antar interface agent dan interfase dari system *telemedicine*, seperti *doctor agent, personal agent*, dan *security agent*. Pada implementation group, diagnosis agent memegang peranan penting, kerena diagnosis merupakan proses yang kompleks, diagnosis *agent* bukan hanya berinterakasi dengan *agent* dalam implementation group, tetapi juga berinterakasi dengan *decision group, clinic group, education agent*, dan *consultation agent,* yang ditunjukkan dengan panah 7. Panah 8 dan 17, merepresentasikan interaksi internal antar *agent* dalam implementation group pada system *telemedicine,* sesuai dengan tabel matrik di bawah ini. Selanjutnya, *traning agent* mengimplementasikan metode pada terapi *agent,* hal ini ditunjukkan pada panah 19.

Dengan melihat uraian diatas terlihat jelas, bahwa penerapan IT dengan

pendekatan AOSE disalah satu bidang kehidupan manusia yaitu dunia kesehatan yang dikenal dengan istilah *e-medicine*, sudah memberikan suatu sistem yang akan dapat bertahan dari kematian, dan dapat menjaga keberlangsungan hidupnya, dengan selalu meningkatkan pemanfaatannya kepada lingkungan dan pengguna. Sistem e-medicine ini dapat mengelola pengetahuan dan pengalamannya, dalam menyelesaikan tugas yang didelegasikan kepadanya, sistem tersebut dapat menerima dan mengenal informasi yang diterimanya, serta melakukan aksi terhadap sesuatu yang diterimanya dari pengguna, lingkungan, maupun dari dalam sistemnya sendiri. Sehingga dapat dikatakan sistem tersebut akan dapat menjaga keberlangsungan hidupnya dan bisa dijamin kualitasnya karena semua kebutuhan-kebutuhan baik dari pengguna, lingkungan maupun sistemnya sendiri dapat terpenuhi.

V. KESIMPULAN

Dari pembahasan diatas dapat ditarik kesimpulan sebagai berikut :

a. Paradigma rekayasa perangkat lunak berorientasi agen (*agent oriented software engineering*, AOSE) merupakan proses pemodelan sistem kompleks dan terdistribusi yang efektif, seperti pengunaan dekomposisi, kesesuaian abstraksi, dan terfokus pada struktur secara organisasional.
b. *Multiagent system engineering* (MaSE) adalah salah satu metodologi AOSE yang digunakan untuk membantu para pengembang dalam merancang dan membangun perangkat lunak berbasis sistem multiagen (*multiagent system*), melalui siklus hidup perangkat lunak dari proses spesifikasi hingga implementasi sistem multiagen.
c. *Multiagent system*, dibangun dari agent-agent yang cerdas (*intelligent agents*) dengan menggunakan konsep *expert system*. Dimana tiap agent memiliki kemampuan dapat menerima informasi dari luar, yang selanjutnya dapat memberikan aksi ke lingkungan, serta dapat menambah pengetahuan dan pengalamannya, sehingga dapat meningkatkan kemampuan dan pemanfaatannya bagi pengguna.
d. Jika perangkat lunak cerdas yang dibangun dengan *multiagent system*, dapat meningkatkan kemampuannya dan menjawab perubahan lingkungan serta memenuhi semun kebutuhan-kebutuhan disekitarnya, maka perangkat lunak tersebut dapat dijamin kualitasnya dan dapat memperpanjang hidupnya.
e. Pembangunan perangkat lunak dengan *multiagent system* telah dilakukan, salah satunya di dunia kesehatan. Yaitu pembangunan telemedicine dengan pendekatan *multiagent system*. Sehingga perangkat lunak dapat ditambah pengetahuannya agar tetap bekerja mengikuti perkembangan pengetahuan disekitarnya.

DAFTAR PUSTAKA

Aan Al Bone,2005 *"Multi-Agent System Sebagai Solusi Pembangunan Perangkat Lunak"*, Proceedings Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005) Yogyakarta

Azhari dan Sri Hartati,2005. *"Overview Metodologi Rekayasa Perangkat Lunak Berorientasi Agen"*, Proceedings Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005) Yogyakarta

Bergenti, F. dan Paola T.,2002, *"Agent-Oriented Software Engineering"*

Jiang, Tian, and Huaglory, Tianfield, 2003, " *A Multiagent Approach to the Design of an E-medicine System*", Journal from School of Computing and Mathematical Sciences Glasgow Caledonian University.

Romi Satria Wahono,2003 *"Pengantar Multi-Agent System (MAS)"*, Copyright © 2003 IlmuKomputer.Com

Tveit. A.,2004, *"A Survey of Agent-Oriented Software Engineering"*, First NTNU CSGSC,2001,www.elcomag.com/amund/ [akses terakhir]

Wooldridge,2002, Michael, *"An Introduction ti Multi-Agent System"*, John Wiley Sons, England.

Wooldridge, M, Jennings N.R dan Kinny, D.,2000, *"The Gaia Methodology for Agent-Oriented Analysis and Design"*, Journal of Autonomous Agents and Multi-agent Systems, 3(3).