Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005)
Yogyakarta, 18 Juni 2005
ISBN: 979-756-061-6

MULTI-AGENT SYSTEM SEBAGAI SOLUSI PEMBANGUNAN PERANGKAT LUNAK DALAM MENJAMIN KEBERLANGSUNGAN HIDUP PERANGKAT LUNAK

Aan Al Bone
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Pasundan
Jl. Setiabudhi No.193 Bandung-40153
E-mail: aanalbone@yahoo.com

Abstrak

Multi-agent system sebagai solusi pembangunan perangkat lunak dalam menjamin keberlangsungan hidup perangkat lunak, menawarkan suatu konsep untuk menghindari berakhirnya kehidupan perangkat lunak lebih cepat. Setiap tugas yang ada, didelegasikan kepada agent, yang secara cerdas mengelola pengetahuannya, dan memberikan kontribusi untuk agent lainnya, maka ketika terjadi perubahan pada lingkungan sekitarnya, hanya agent yang terkaitlah yang ditambah atau diubah pengetahuan dan kerjanya, tanpa harus mengganggu agent lainnya. Sehingga keberlangsungan hidup perangkat lunak akan lebih panjang.

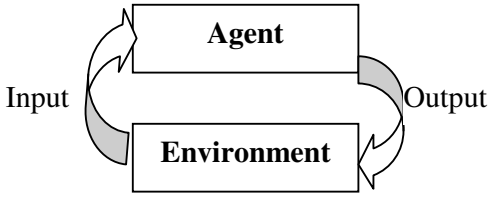
Kata Kunci: *Agent, Multi-Agent System, Intelligent Agent, Expert System*

1. Pendahuluan

Beberapa trend yang menandai perkembangan perangkat lunak, antara lain *interconnection, intelligence,* dan *delegation*. *Interconnection* berarti sistem komputer yang saat ini tidak lagi *stand alone*, tetapi telah berkembang menjadi sistem yang terdisitribusi. Sedangkan *intelligence* berarti semakin kompleksnya kerja yang dilakukan oleh perangkat lunak, sehingga kemudian perangkat lunak didesain untuk bekerja secara *automatic*. Adapun *delegation* berarti perangkat lunak akan menerima delegasi pekerjaan, yang sesuai dengan fungsi yang diinginkan. Hal-hal yang disebutkan diatas, bagaikan bola salju yang mengarah kepada munculnya *multi-agent system.*

2. *Intelligent Agents*

Intelligent Agent adalah sebuah sistem komputer yang *independent*, yang mampu melakukan *action* terhadap kebutuhan lingkungannya, khususnya kebutuhan dirinya sendiri, sehingga sebuah *agent* dapat menyelesaikan pekerjaan yang didelegasikan kepadanya secara lebih baik (*delegation*), serta mampu secara cerdas mengelola pengetahuannya (*intelligence*), juga memberikan kontribusi pada lingkungannya (*interconnection*), seperti gambarkan dibawah ini.(2)



Gambar 1. Interaksi Agent dan Lingkungannya

Agent yang cerdas atau *Intelligent agents* merupakan sistem komputer yang reaktif (*reaktive*), pro-aktif (*pro-active*) dan sosial (*social*) terhadap lingkungannya.

Reaktif (*reaktive*) berguna untuk memelihara interaksi dengan lingkungannya, dan merespon perubahan yang terjadi. Sedangkan pro-aktif berbeda dengan reaktif, pro-aktif (*proactiveness*) artinya berusaha untuk mencapai tujuannya dengan berinisiatif, tanpa menunggu event atau kejadian tertentu. Sedangkan kemampuan sosial (*social ability*) berarti agent akan selalu berhubungan dengan agent lainnya. Karena dalam mencapai tujuan atau tugasnya, agent harus saling mendukung.

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.

ISBN: 979-756-061-6

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 (3):

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:

a. *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.
b. *Forward Chaining*, dengan pendekatan yang berbeda dari backward chaining, forward chaining melakukan pencarian dari suatu masalah kepada solusinya.

4. *Agent* dan *Multi-Agent System*

Selanjutnya kita mengenal lebih dalam tentang konsep agent dan multi-agent itu sendiri.

4.1 Agent

Agent harus memiliki kemampuan untuk mengenal dan menerima infomasi dari luar (*perception*), dan selanjutnya melakukan aksi (*action*).(5)

a. Perception

Manusia menerima sesuatu dari lingkungannya, dengan melihat, mendengar atau dengan rasa. Apa yang diterimanya melalui proses filter, sehingga dapat juga menolak sesuatu.

Intelligent Agent harus memiliki juga kemampuan menerima informasi dari lingkungannya. Informasi itu bisa terimanya dengan menggunakan sensor. Intelligent Agent juga dapat mengirimkan pesan permintaan informasi ke agent lainnya. Agent harus dapat melihat dengan baik normal event (*mouse movement)* dari significant event (*double click*). Jika Intellligent Agent bekerja untuk memonitor *email* atau *newsgroup*, agent tersebut harus dapat mengetahui jika menerima dokumen baru. Kemampuan untuk memperhatikan atau mengenal informasi yang tersembunyi dalam data merupakan hal yang sulit. Karena

ISBN: 979-756-061-6

membutuhkan *intelligence* dan pengetahuan. Hal ini lah yang disebut dengan "*perceptive*".

b. Action

Setelah agent tersebut memiliki kemampuan perceptive dalam mengenal informasi yang diterimanya, agent harus dapat mengambil *action*. *Action* yang dilakukan bisa saja dengan tidak bereaksi atau bisa dengan mengirimkan pesan ke agent lainnya. Seperti manusia yang bereaksi dengan cara berbicara atau mengirimkan email untuk berkomunikasi dengan orang-orang yang ada di lingkungannya.

4.2 Multi-Agent System

Dari pembahasan diatas bisa terlihat bagaimana intelligent agent dapat membantu kerja manusia. Jika banyak hal yang harus ditangani dalam sebuah sistem, maka dibutuhkan banyak orang yang mengerjakannya. Begitupun dengan agent. Semakin luas sistem yang ditangani, maka dibutuhkan beberapa agent. Hal inilah yang disebut "*Multi-agent System*".

Ketika satu agent ingin berkomunikasi dengan agent lainnya, dapat dilakukan dengan berbicara langsung, tentunya dengan bahasa yang sama. Terdapat level bahasa dasar agent, pertama *syntax* dan format pesan, kemudian yang kedua adalah *meaning and semantics*. Ketika sintaks sulit dipahami, tetapi semantics tidak. Contohnya ketika dua orang saling bercakap-cakap dan saling tidak memahami antar satu dengan lainnya, kerena yang satu membicara tentang kapal laut, sedangkan seorang lainnya berbicara tentang kapal terbang. Maka diperlukan *shared vocabulary* dan artinya. *Shared vocabulary* itu disebut *ontology*. Tiap bidang dan hal memiliki *ontology,* contohnya *automobile ontology*, *computer ontology* dan lain-lain.(1)

5. Pembangunan *Agent* dan *Multi-Agent System*

Agent dikembangkan dibidang industri dan kesehatan. Terdapat dua standarisasi yang ditetapkan dalam pembangunan Agent. Pertama adalah The *Foundation for Intelligent Physical Agents* (FIPA), dan yang kedua adalah *the Object Management* Group (OMG).

5.1 *Foundation for Intelligent Physical Agents* (FIPA)

FIPA memiliki paltform, yang terdiri dari agent management, agent communication dan agent software integration. Agent management menangani agent life cycle, directory facilitator, dan message transport system. Sedangkan agent communication menangani protokol yang digunakan dalam interaksi antar agent. Sedangkan agent software integration, menangani integrasi agent dalam sebuah sistem.

5.2 *Object Management Group* (OMG)

Standard OMG, banyak digunakan dalam *mobile agent system*.

6. Intelligent Agent Framework

Pembangunan intelligent agent framework menggunakan *object oriented* desain. Dimulai dengan mendefinisikan spesifikasi kebutuhan (*Requirement*), kemudian membuat desain *goal* dan *functional specification.*

6.1 Requirement

Langkah pertama adalah dengan mengumpulkan permintaan kebutuhan dari komunitas user. Kebutuhan lainnya yang harus diperoleh adalah prinsip dasar yang akan digunakan untuk dapat memecahkan masalah.

6.2 Desain Goals

Spesifikasi kebutuhan yang datang dari user, dapat menjelaskan function dan property dari produk yang akan dibuat.

6.3 Functional Specification

Setelah membuat *requirement* dan desain *goals,* langkah selanjutnya diturunkan menjadi *functional specification*. *Functional specification* akan menghubungkan antara *development team* dan *user community*. Contoh *functionality* yang dibutuhkan, antara lain:

a. Agent harus mendukung proses yang ada dalam sistem.
b. Harus dapat diberikan tambahan pengetahuan kepada agent.

Dengan dibangunnya *Intelligent Agent framework*, maka dapat didesain Multi-Agent yang mendukung suatu sistem. Dimana multi-agent system terdiri dari agent-agent yang cerdas (*Intelligent Agents*), yang dapat mengelola pengetahuan dan pengalamannya, untuk dapat memberikan solusi dan reaksi terhadap lingkungannya, sehingga delegasi tugas yang diberikan dapat dikerjakan secara cepat dan baik.

7. Keberlangsungan Hidup Perangkat Lunak

Tidak ada yang abdi di dunia ini. Itulah yang diajarkan oleh agama kepada manusia. Tetapi bukan berarti manusia dilarang berusaha untuk mempertahankan hidup. Karena sesungguhnya manusia telah diberikan akal oleh Allah SWT, untuk berusaha mencari reski dan berpikir demi mempertahankan hidupnya. Manusia mempertahan-kan hidupnya untuk dapat memberikan manfaat

ISBN: 979-756-061-6

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 yang didelegasikan 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*).

8. Contoh Pembangunan Perangkat Lunak dengan *Multi-Agent System* di Bidang Kesehatan

Dalam dunia kesehatan dikenal e-medicine. E-medicine adalah penerapan IT (*Information Technology*) dalam dunia kesehatan. E-medicine meliputi *telemedicine*, *e-healthcare, e-diagnosis, e-consultation, e-clinic*, dan lain-lain. Dalam hal ini, kita akan membahas pembangunan telemedicine dengan pendekatan *Multi-Agent System.*

Pendekatan *Multi-Agent System* dalam pembangunan telemedicine, dilakukan dengan langkah-langkah sebagai berikut (4):

1. *Requirement Analysis*
2. Desain
   c. Desain *Goals* and *Functional Specification*
   d. Desain Agent
   e. Desain *Multi Agent Society*

8.1 Requirement Analysis

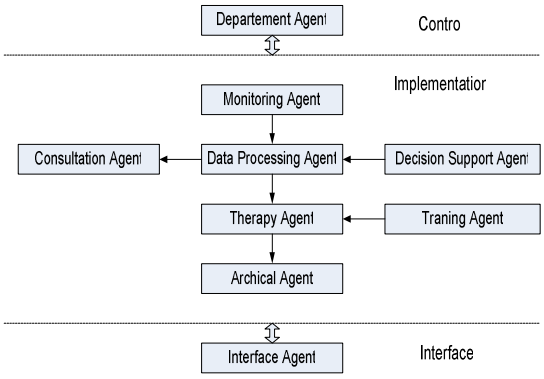
Sistem telemedicine harus dapat memberi pemantauan dan pelayanan setiap saat kepada pasien. Adapun analisis kebutuhan, sebagai berikut:

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 penyulusah 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

8.2 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 2. Arsitektur Multi Agent System dati 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.

8.3 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,*

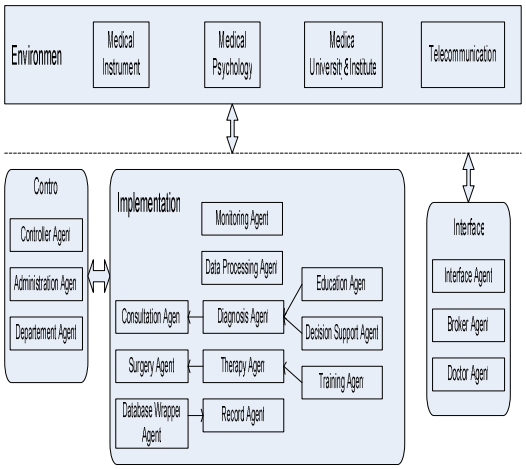
 ISBN: 979-756-061-6

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.

8.4 Desain *Multi-Agent System Society*

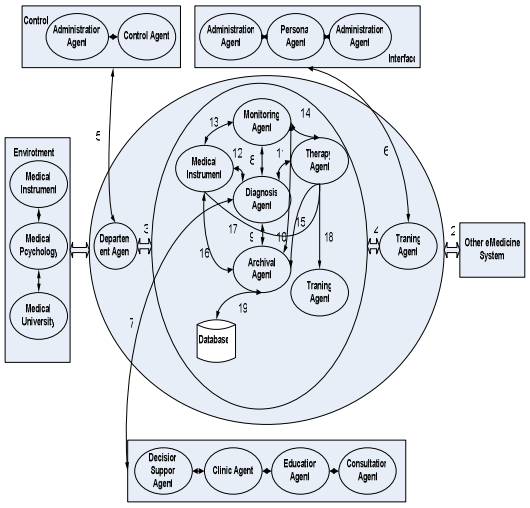
Desain *multi agent society* fokus pada pendirian arsitektur sistem *multi agent* dan interaksi antar *agent*. Model interaksi dalam sistem *multi agent* dibagi menjadi model eksternal dan model internal. Model interaksi internal dari sistem *telemedicine* bukan hanya antara agent dengan satu departemen,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



Gambar 3. Multi-Agent Society

Pertama, terlebih dahulu harus diketahui informasi-informasi yang dibutuhkan untuk perencanaan dan pencapaian tujuan dalam interaksi antar *agent.* Dalam sistem *multi agent,* hubungan eksternal fokus pada upaya intergrasi bagian-bagian *telemedicine* dan lingkungannya. Sedangkan hubungan internal fokus pada kerjasama antar agent untuk merealisasikan terjadinya interakasi tersebut.

Dibawah ini gambaran interakasi yang terjadi pada sistem muli agent di *telemedicine* diabetes.



Gambar 4. Interaksi pada Multi-Agent System

Panah 1 dan 2, menggambarkan bahwa telemedicine lingkungan di luarnya dan sistem 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 *departement agent* dan *control group* dalam *telemedicine*, seperti

ISBN: 979-756-061-6

interaksi antara departemen administrasi dan control group. Panah 6, merepresentasikan interaksi antar interface agent dan interfase dari sistem *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 sistem *telemedicine,* sesuai dengan tabel matrik di bawah ini. Selanjutnya, *traning agent* mengimplementasikan metode pada terapi *agent,* hal ini ditunjukkan pada panah 19.

9. Kesimpulan

Dari penjelas diatas, maka dapat disimpulkan, sebegai berikut:

a. Multi-Agent system, dibangun dari agent-agent yang cerdas (*Intelligent Agents*). Dimana tiap agent memiliki kemampuan dapat menerima informasi dari luar, yang selanjutnya dapat memberikan aksi ke lingkungan.
b. *Intelligent agents* dapat menambah pengetahuan dan pengalamannya, sehingga dapat meningkatkan kemampuan dan pemanfaatannya bagi pengguna.
c. *Intelligent Agent* dalam mengelola pengetahuannya, menggunakan konsep *expert system* untuk mentransfer pengetahuan dari luar lingkungannya, proses tersebut meliputi *knowledge acuisition, knowledge representation, knowledge inferecing dan knowledge transfering*
d. Jika perangkat lunak cerdas yang dibangun dengan Multi-Agent System, dapat meningkatkan kemampuannya dan menjawab perubahan lingkungan sekitarnya, maka perangkat lunak dapat memperpanjang hidupnya.
e. Pembangunan perangkat lunak dengan *multi-agent system* telah dilakukan di dunia kesehatan. Yaitu pembangunan telemedicine dengan pendekatan *multi-agent system*. Sehingga perangkat lunak dapat ditambah pengetahuannya agar tetap bekerja mengikuti perkembangan pengetahuan disekitarnya.

Referensi

[1] Bayegan, Elisabeth, and Nytrø,Oystein, and Grimsmo, Anders, "*Ontologies for Knowledge Representation in a Computer-Based Patient Record*", Journal from Department of Computer and Information Science Department of Community Medicine and General Practic, 1999.
[2] Bigus, Joseph, and Bigus, Jennifer, "*Constructing Intellegent Agents Using Java*", John Wiley Sons, England, 2002.
[3] Herman Tolle, ST., MT. "Materi Kuliah Sistem Pakar".
[4] Jiang, Tian, and Huaglory, Tianfield, " *A Multi agent Approach to the Design of an E-medicine System*", Journal from School of Computing and Mathematical Sciences Glasgow Caledonian University, 2003.
[5] Wooldridge, Michael, "*An Introduction ti Multi-Agent System*", John Wiley Sons, England, 2002.