

SISTEM AGEN CERDAS BERBASIS WEB UNTUK KOMPUTASI INFORMASI KINERJA DARI PELAKSANAAN PROYEK-PROYEK PERUSAHAAN

Azhari Subanar Retantyo Wardoyo Sri Hartati

Ilmu Komputer, FMIPA, UGM

Email: arisn@ugm.ac.id, subanar@yahoo.com, rw@ugm.ac.id, shartati@ugm.ac.id

ABSTRACT In this paper are presented the development of a model of intelligent software agents to support web-based computing of project performance information that being executed by the company. The project performance is calculated by using earned value management method. We have developed a prototype of the intelligent agent software based on Jadex agent framework. Based on some simple data simulations of enterprise projects, our agents have showed capabilities in calculating the performance value of projects and displaying into barchart or S-graph.

Keywords: Intelligent software agent, multiagent system, enterprise projects, project performances, earned value management.

ABSTRAK Dalam makalah ini disajikan pengembangan model perangkat lunak agen cerdas untuk mendukung komputasi berbasis web informasi kinerja proyek yang sedang dilaksanakan oleh perusahaan. Kinerja proyek dihitung dengan menggunakan metode manajemen nilai hasil. Kami telah mengembangkan sebuah prototipe perangkat lunak agen cerdas yang didasarkan pada kerangka agen Jadex. Didasarkan pada beberapa data sederhana simulasi proyek-proyek perusahaan, agen kami telah menunjukkan kemampuan dalam menghitung nilai kinerja proyek dan menampilkan ke barchart atau S-grafik.

Kata Kunci: Perangkat lunak agen cerdas, sistem multi agen, proyek enterprise, performansi proyek, earned value management.

Penelitian dan penerapan pendekatan sistem multiagen cerdas pada bidang sistem informasi perusahaan telah mulai banyak diminati dan berkembang sangat pesat dalam beberapa tahun terakhir [1]. Termasuk terhadap pengembangan aplikasi cerdas untuk proses-proses atau kegiatan-kegiatan dalam area manajemen proyek-proyek perusahaan [2], [3], [4]. Sebuah sistem agen cerdas secara tunggal atau sistem dengan beberapa agen-agen dapat dibangun untuk membantu mengefektifkan kegiatan-kegiatan proyek mulai dari penjadwalan [5], monitoring kontrak, hingga kolaborasi antara tim proyek dan bisnis [6], [7].

Secara prinsipil, menurut Mohammadian[3] dan Wooldridge [8] terdapat beberapa kecenderungan yang memacu pemanfaatan dari model komputasi berbasis kecerdasan agen saat ini, diantaranya karena (1) *ubiquity*, komputer terdapat dimana-mana, menyebar diberbagai lokasi; (2) *interconnection*, komputer dapat saling terhubung satu sama lainnya; (3) *intelligence*, komputer dapat melakukan aktivitas-aktivitas kompleks dan penalaran secara otomatis; (4) *delegation*, manusia dapat mendelegasikan sebagian atau banyak pekerjaannya kepada komputer, misalnya agar komputer dapat berfungsi sebagai pemegang kendali penerbangan pesawat; (5) *human-oriented*, komputer dapat bekerja seperti cara kerja manusia, misalnya melakukan kerja sama dan koordinasi.

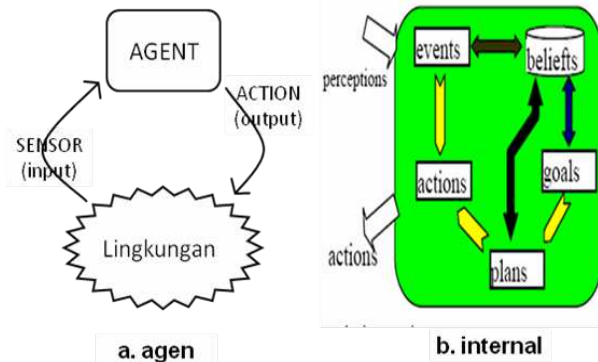
Sebuah sistem ataupun aplikasi yang dibangun dengan pendekatan model kecerdasan multiagen terhadap kondisi tersebut ternyata mampu menunjukkan peningkatan kinerja atau kemampuan sistem yang luar biasa. Sistem informasi menjadi juga lebih baik, mudah, efisien dan efektif jika dibandingkan jika dibangun dengan pendekatan metodologi tradisional seperti pemrograman prosedural, predi-

kat, ataupun pemrograman berbasis-obyek [6].

Goethert [9] dan Marchewka[10], menyebutkan bahwa dalam bidang manajemen proyek, salah satu kunci sukses dari setiap direktur proyek ataupun manajer proyek perusahaan untuk mengendalikan proyek-proyek adalah melalui monitoring, mengukur, dan mengelola kinerja setiap proyek perusahaan. Informasi ini dapat digunakan baik untuk kepentingan kebijakan internal manajemen perusahaan, maupun untuk eksternal seperti meningkatkan kualitas pelayanan, kolaborasi, dan komunikasi perusahaan terhadap semua pihak yang saling terkait dalam pelaksanaan proyek. Seperti pelaksana proyek, penyandang dana, stakeholder, kolega, pemilik atau pelanggan.

Satu persoalan menarik yang cukup mendapat perhatian dari pihak manajemen perusahaan sejak beberapa tahun ini adalah belum tersedianya perangkat lunak komputasi cerdas yang fleksibel dan mampu menyediakan data dan informasi kinerja dari pelaksanaan dan perkembangan proyek-proyek perusahaan. Perangkat lunak aplikasi ini seharusnya dapat menyediakan berbagai informasi kemajuan pelaksanaan proyek terkini secara terus menerus dan otomatis, serta terintegrasi dengan sistem perusahaan lainnya. Banyak aplikasi-aplikasi monitoring, evaluasi dan pengendalian proyek telah mulai dikembangkan dengan mengintegrasikan paradigma kecerdasan buatan. Namun, pada umumnya kendala utama yang muncul adalah masih sulit dan kurang efektif digunakan secara bersama-sama dalam lingkup operasional manajemen yang lebih kompleks dan luas. Di samping itu, banyak pula aplikasi agen cerdas yang ada belum terintegrasi ke dalam teknologi internet.

Dalam makalah ini dibahas pengembangan sebuah model perangkat lunak agen cerdas berbasis web untuk mendukung



Gambar 1: Abstraksi sebuah agen cerdas dan lingkungannya

proses komputasi informasi kinerja pelaksanaan dari proyek-proyek perusahaan yang sedang dilaksanakan. Melalui paradigma sistem multi agen, memungkinkan aplikasi dibangun memiliki kemampuan agen-agen cerdas seperti dalam dunia nyata. Misalnya kemampuan mandiri, proaktif, sosial, berinteraksi, bernegosiasi terhadap sesama agen cerdas dalam lingkungannya, ataupun terhadap para pemakai sistem untuk melakukan komputasi kinerja dari pelaksanaan proyek.

PEKERJAAN RELEVAN

Berbagai pendekatan dan model untuk membangun sistem aplikasi pengelolaan proyek-proyek perusahaan terus dilakukan dan dikembangkan dari berbagai aspek baik fungsional maupun untuk prioritas tertentu. Seperti pendekatan model sistem klasik berbasis informasi manajemen, pendekatan model sistem pakar dan konsultasi proyek, termasuk juga pendekatan kecerdasan buatan masih sering dilakukan, seperti yang dapat dilihat pada makalah Danilovic dan Sandkull [11] dan Bergamaschi, et al [2]. Sedangkan Pitt, et al [12] dalam makalahnya telah pula mengusulkan sebuah kerangka formal penerapan sistem agen untuk persoalan kompleks manajemen proyek antar organisasi, melalui pemanfaatan kemampuan protokol komunikasi internet. Penelitian dan penerapan sistem agen untuk penjadwalan proyek telah pula diteliti oleh Yan, et al [4]. Namun penelitian tersebut belum mengintegrasikan dengan fasilitas monitoring dan pengendalian proyek-proyek.

Penelitian dan penerapan sistem agen untuk mendukung manajemen juga telah mulai dilakukan seperti dalam makalah El-Khouly [13], Chaves, et al [14], Pitt, et al [15] dan Qomariah [16]. Makalah-makalah tersebut menyebutkan bahwa pendekatan sistem agen semakin luas digunakan, karena mampu meningkatkan kinerja sistem aplikasi yang dikembangkan. Chaves, et al [17], mencoba membuat atau menerapkan metode multiagen untuk mengelola alokasi sumberdaya komputer yang tersebar, yang diberi nama dengan Challenger. Disini, agen-agen dirancang bekerja bersama-sama untuk mengelola sumberdaya lokal dengan tujuan menyeimbangkan pemakaian seluruh prosesor, agen-agen berkomunikasi melalui pengiriman status pemanfaatan prosesor untuk dapat dimanfaatkan. Dengan demikian diharapkan mampu meminimalkan rata-rata waktu untuk melakukan pekerjaan-pekerjaan pelayanan dari sistem.

Meskipun agen-agen tidak dirancang untuk proses manajemen perusahaan, namun prinsip komunikasi atau model kerjasama agen-agen dalam lingkungan tersebar dapat dikembangkan untuk melakukan proses monitoring proyek dan menganalisis rekomendasi pengendalian manajemen seperti penelitian yang akan diusulkan ini. Sedangkan dalam Qomariah [16], model interaksi agen cerdas telah digunakan untuk pengendalian persetujuan pengeluaran biaya proyek. Model interaksi agen yang dikerjakan hanya dibangun untuk proses proyek tunggal dan belum mendukung untuk memfasilitasi para stakeholder proyek. Sedangkan model yang diusulkan pada penelitian ini, model kolaborasi agen-agen untuk banyak proyek.

Dalam manajemen proyek, kemampuan berinteraksi antar agen-agen cerdas dapat pula dimanfaatkan untuk melakukan kegiatan negosiasi kesepakatan harga atau waktu proyek. Kemampuan lainnya yang dapat diciptakan adalah agen untuk mampu bekerja sesama agen [18] baik dalam komunitas dengan jenis agen-agen yang sama (dalam sistem multiagen homogen) atau dengan komunitas dengan sistem multiagen yang berbeda lainnya (sistem multiagen heterogen). Sebagai contoh penerapan pada bagian produksi dan penjualan yang dilakukan oleh Gerber dan Klusch [19]. Kemampuan bekerja sama antar agen-agen ini sangat potensial untuk diterapkan terhadap persoalan kompleks dari proses-proses manajemen proyek-proyek perusahaan yang lebih global, tersebar, dan menyeluruh [17].

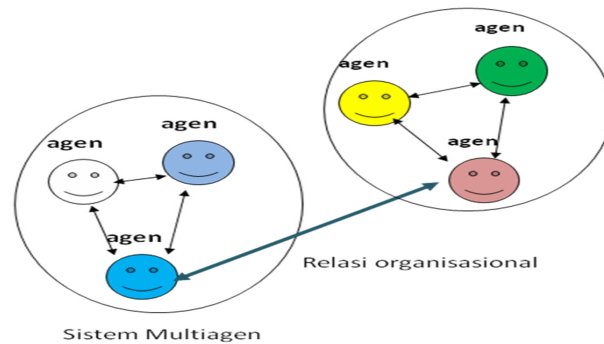
PERANGKAT LUNAK AGEN CERDAS

Agen Cerdas

Pada makalah ini, definisi sebuah agen disadur dari Wooldridge [8], yang menyatakan bahwa sebuah agen adalah sebuah sistem komputer yang berada dalam suatu lingkungan dan memiliki kemampuan bertindak secara otonom didalam situasi lingkungan tersebut sesuai dengan sasaran yang dirancang.

"An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives".

Pada Gambar 1a, diperlihatkan abstraksi dari model komputasi sebuah agen. Pada gambar terlihat setiap tindakan atau aktivitas akan dikerjakan oleh agen adalah untuk memenuhi kondisi lingkungannya. Sedangkan pada Gambar 1b, diperlihatkan komponen internal dari sebuah



Gambar 2: Sistem Multiagen

model agen BDI (belief-desire-intention) yang memiliki events (pemacu indera), beliefs (pengahuan), actions (tindakan), goals (tujuan), dan plans (agenda dan rencana).

Menurut Russel dan Norvig [20], sebuah agen adalah segala sesuatu yang dapat merasakan lingkungannya melalui peralatan sensor-sensor, bertindak sesuai dengan lingkungannya dan dengan menggunakan peralatan penggerak (actuator). Sebagai contoh termasuk manusia, robotika, atau perangkat lunak agen.

Sebuah agen selalu mencoba untuk mengoptimalkan sebuah nilai ukuran kinerja yang disebut agen memiliki rasional (*rational agent*). Sebuah agen adalah rasional jika dapat memilih kemungkinan untuk bertindak yang terbaik setiap saat, menurut apa yang ia ketahui mengenai lingkungannya pada saat itu. Ukuran kinerja (dari *rational agent*) biasanya didefinisikan oleh perancang agen dan merepleksikan apa yang diharapkan mampu dilakukan dari agen tersebut. sebuah agen berbasis rasional juga disebut sebuah agen cerdas [6].

Melihat definisi rasional agen tersebut, secara umum dapat dikatakan mirip seperti pada manusia (mata sebagai sensor, tangan sebagai penggerak), agen-agen robotika (kamera sebagai sensor, roda sebagai penggerak), perangkat lunak (antarmuka pengguna grafis sebagai sensor dan sekaligus sebagai penggerak). Dari perspektif ini, bidang kecerdasan buatan dapat dipandang sebagai studi mengenai prinsip-prinsip dan perancangan dari rasio agen-agen buatan [3] [12].

Sistem Multiagen

Sedangkan model komputasi sebuah sistem multiagen, menurut Sycara [12] dapat dipandang sebagai sebuah agen yang termasuk dalam sekumpulan agen-agen lainnya dan saling berinteraksi satu agen dengan agen lainnya, untuk mencapai sasaran yang dirancang.

Berdasarkan pada pengertian tersebut sebuah sistem perangkat lunak kompleks dapat dinyatakan sebagai sebuah koleksi dari banyak agen-agen autonomous berukuran kecil, masing-masing memiliki fungsi-fungsi dan sifat-sifat sendiri, dan berinteraksi diantara agen-agen melakukan integrasi sistem secara menyeluruh.

Pada Gambar 2, diperlihatkan skema abstraksi dari model komputasi sistem multiagen. Di dalam setiap subsistem agen-agen berinteraksi dalam lingkungannya, sedangkan dengan subsistem yang lain berinteraksi dalam bentuk

relasi organisasional [21] [12].

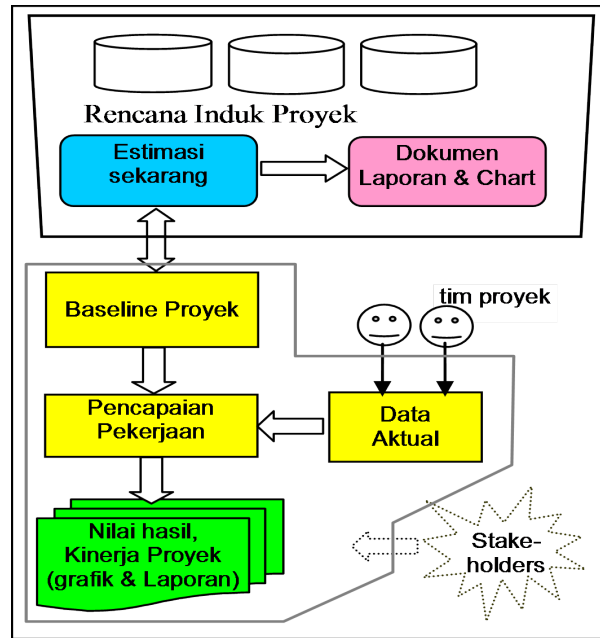
Wooldridge [8] menetapkan tujuan utama penelitian dalam bidang komputasi sistem multiagen adalah: pertama, Agen-agen sebagai paradigma untuk rekayasa perangkat lunak. Para ahli bidang rekayasa perangkat lunak telah maju selangkah dalam memahami karakteristik-karakteristik dari kompleksitas perangkat lunak. Dalam pengertian interaksi merupakan satu karakteristik yang penting dari model komputasi kompleks. Kedua, Agen-agen sebagai sebuah alat untuk memahami model relasi sosial antar manusia. Sistem multiagen menyediakan sebuah cara baru untuk melakukan simulasi model komputasi hubungan sosial kemasyarakatan, yang mungkin akan membantu membuka pemahaman dari berbagai jenis proses interaksi sosial.

INFORMASI KINERJA KEMAJUAN PROYEK

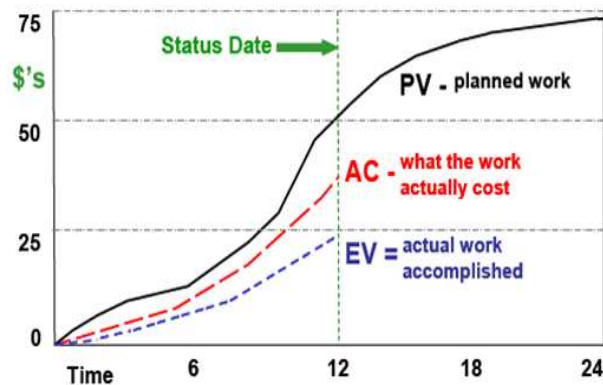
Kegiatan monitoring pelaksanaan proyek pada umumnya meliputi pekerjaan-pekerjaan melakukan pelacakan kemajuan, menyediakan sejumlah informasi perbandingan antara data pelaksanaan aktual proyek terhadap data prediksi awal menurut baseline, melakukan analisis dampak pelaksanaan proyek, dan membuat penyesuaian atau koreksi terhadap setiap penyimpangan pelaksanaan proyek.

Dalam konteks aplikasi, informasi monitoring untuk proyek-proyek perusahaan sebaiknya harus tersedia secara online, dan realtime agar mudah diakses oleh semua pihak yang berkepentingan terhadap proyek. Pada gambar 3, diperlihatkan alur proses perhitungan nilai-nilai hasil atau informasi kinerja proyek dari sebuah kegiatan monitoring proyek. Ketersediaan informasi monitoring proyek secara online, cepat, akurat dapat membantu meningkatkan kinerja komunikasi terhadap stakeholders. Selanjutnya hasil pelacakan dan informasi kinerja monitoring proyek dapat disampaikan dalam bentuk-bentuk laporan resmi, tekstual atau grafik melalui laporan resmi, email, rapat resmi, pertukaran data/file, ataupun dalam bentuk web [22].

Metode manajemen nilai hasil (Earned Value), dan nilai estimasi beban sisa pekerjaan hingga saat selesai dipandang cara yang paling mudah untuk melakukan monitoring maupun pengukuran kinerja kemajuan dari pelaksanaan proyek. Stratton [23] dan Lukas [24] menuliskan bahwa nilai kinerja proyek dalam EVM (Earned Value Management) pada dasarnya menggunakan tiga aspek informasi dari proyek, yaitu nilai yang direncanakan (Plan Value, PV), biaya aktual (Actual Cost, AC) dan nilai hasil atau ni-



Gambar 3: Alur informasi kinerja kemajuan proyek



Gambar 4: Kurva S untuk nilai PV, AC dan EV

lai yang diperoleh (Earned Value, EV). Nilai-nilai kinerja tersebut dapat dihitung dalam persen dan digambarkan dalam sebuah kurva S, misalnya seperti yang ditampilkan dalam Gambar 4. Planned Value (PV), adalah anggaran biaya untuk pekerjaan yang dijadwalkan dan harus dilakukan. Ini adalah bagian dari rencana anggaran proyek yang akan dikeluarkan pada suatu titik waktu. Nilai ini juga dikenal sebagai Budgeted Cost of Work Scheduled (BCWS).

Actual Costs (AC), adalah dana yang telah dihabiskan untuk pekerjaan yang sedang dilakukan dan telah dicapai. Nilai ini juga dikenal sebagai Actual Cost of Work Performed (ACWP).

Earned Value (EV), adalah persen dari total anggaran aktual yang telah dicapai pada suatu titik waktu tertentu. Nilai ini juga dikenal sebagai Budgeted Cost of Work Performed (BCWP). EV dihitung dari perkalian rencana anggaran untuk suatu kegiatan dengan persentase kemajuan yang dicapai untuk kegiatan tersebut, atau $EV = (\% \text{ se-}$

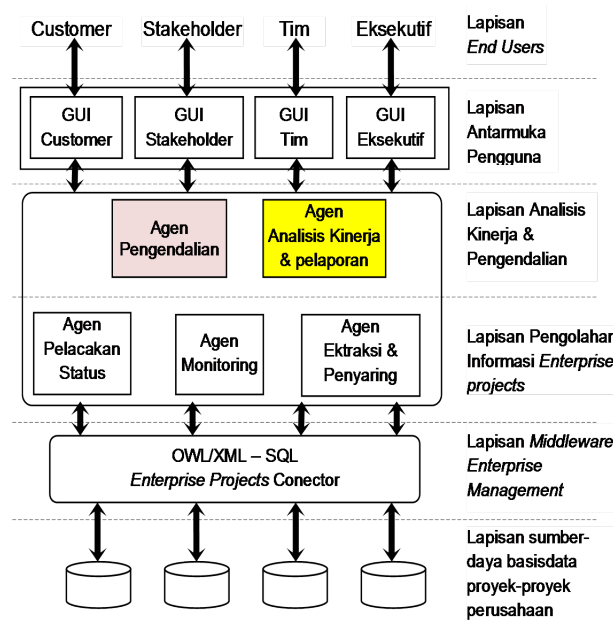
lesai) * rencana anggaran, atau rumus (1), (2).

$$EV = QtyInstalled \times EstimatedUnitRate \quad (1)$$

$$EV = BudgetedHours \times \%Complete \quad (2)$$

Masih menurut Stratton [23] dan Lukas [24], berdasarkan nilai PV, EV dan AC, beberapa nilai perhitungan kinerja turunannya dapat mudah dilakukan yang memberikan informasi penting tentang bagaimana proyek lakukan. Misalnya deviasi jadwal, deviasi biaya, indeks kinerja biaya, dan indeks kinerja jadwal. Jika indeks kinerja kurang dari satu berarti pengeluaran lebih besar dari anggaran atau waktu pelaksanaan lebih lama dari jadwal yang direncanakan, dan sebaliknya jika angka indeks kinerja lebih dari satu maka kinerja penyelenggara proyek lebih baik dari rencana, yaitu pengeluaran lebih kecil, jadwal lebih cepat. Rumus untuk menghitung nilai-nilai kinerja tersebut adalah:

$$CostVariance, CV = EV - AC \quad (3)$$



Gambar 5: Arsitektur sistem multiagen untuk kinerja kemajuan proyek-proyek

$$ScheduleVariance, SV = EV - PV \quad (4)$$

$$CostPerformanceIndex, CPI = EV/AC \quad (5)$$

$$SchedulePerformanceIndex, SPI = EV/PV \quad (6)$$

Indeks kinerja biaya (CPI) adalah indikator yang sangat baik untuk melihat efisiensi biaya dari penyelesaian pekerjaan proyek. Penggunaan penting dari CPI adalah untuk peramalan biaya akhir proyek. Beberapa istilah yang digunakan terkait dengan nilai estimasi biaya penyelesaian proyek adalah: Budget at Completion (BAC), jumlah total anggaran yang disetujui apabila lingkup proyek selesai (termasuk proyek kontinjensi). Estimate to Complete (ETC), perkiraan biaya tambahan yang dibutuhkan untuk menyelesaikan proyek. Dan, Estimate at Completion (EAC), perkiraan total sisa biaya yang diharapkan jika proyek akan diselesaikan. Hubungan nilai-nilai BAC, ETC, dan ETC. Nilai ramalan dari EAC dapat dihitung dengan rumus-rumus berikut:

$$EAC = AC + (BAC - EV) \quad (7)$$

$$EAC = BAC/CPI \quad (8)$$

$$EAC = BAC/(CPI * SPI) \quad (9)$$

Rumus (7) menganggap rencana akan dicapai untuk sisa pekerjaan ($CPI = 1.0$), dan hasil yang paling optimis EAC bila proyek tidak bekerja dengan baik. Rumus (8) disebut sebagai 'kumulatif CPI', dan menganggap seluruh proyek akan dilakukan pada kinerja biaya yang sama (atau CPI tidak berubah). Rumus (9) menganggap baik biaya dan jadwal berdampak pada EAC, dan perhitungan yang paling pesimis EAC adalah jika proyek tidak bekerja dengan baik.

ARSITEKTUR SISTEM

Pada penelitian ini, model perangkat lunak yang dibutuhkan diasumsikan mampu menyediakan akses dan proses terhadap data dan informasi perkembangan proyek dari berbagai lokasi server proyek departemen yang terhubung dalam jaringan komputer perusahaan. Data dan informasi proyek disimpan dalam bentuk basisdata dan dikelola oleh sekumpulan agen-agen cerdas. Agen-agen cerdas ini harus mampu merespon terhadap permintaan data dan informasi kinerja proyek dari pengguna, seperti persentase perkembangan proyek dan detailnya, laporan pemanfaatan sumberdaya proyek, indikator kinerja perkembangan proyek dalam bentuk grafik bar chart dan kurva S.

Pada Gambar 5, diperlihatkan arsitektur model multi-agen ini secara menyeluruh. Pada gambar terlihat ada beberapa agen cerdas yang diusulkan saling bekerja sama untuk mendukung proses dan analisis kinerja kemajuan dari pelaksanaan proyek-proyek. Agen-agen tersebut dibedakan dan berkerja pada beberapa lapisan yang memperlihatkan struktur model dari aplikasi. Yaitu (i) agen-agen pada lapisan antar muka, yang bertujuan untuk mengelola tampilan antarmuka dan dialog dalam bentuk web window, (ii) agen-agen pada lapisan analisis kinerja dan pengendalian, (iii) agen-agen pada lapisan pengolahan informasi dan basis data proyek, serta agen yang bekerja pada lapisan middleware yang berkerja sebagai agen pengatur dan pengendali koneksi terhadap server basisdata bagi setiap agen-agen yang memerlukan akses data.

Agen-ekstraktor-penyaring, bekerja sebagai agen penye-rap dan penyaring data dan informasi proyek-proyek dari data induk proyek yang menyimpan atau tersedia dalam setiap server unit kerja perusahaan dan kemudian menyimpan menjadi dalam bentuk dokumen referensi proyek. Data utama setiap proyek yang diserap adalah berupa kegiatan utama dengan jadwal, biaya, milestone, para pelaksana).

Agen-pelacak-status, bekerja sebagai agen pengumpu-

```

<agent>
<goals>
  <achievegoal name="gShowProgress">
    <parameter name="client" class="Socket"/>
  </achievegoal>
  <achievegoal name="gShowProjectPerformance">
    <parameter name="client" class="Socket"/>
  </achievegoal>
  <achievegoal name="gDisplaySGraph">
    <parameter name="client" class="Socket"/>
  </achievegoal>
  <achievegoal name="gDetailGanttProjects">
    <parameter name="client" class="Socket"/>
  </achievegoal>
  <achievegoal name="gDisplayReportStaff">
    <parameter name="client" class="Socket"/>
  </achievegoal>
</goals>
<plans>
  <plan name="pShowProjectProgress">
    <parameter name="client" class="Socket">
      <goalmapping ref="gShowProgress.client"/>
    </parameter>
    <body class="MonitoringProgressProjectPlan"/>
    <trigger>
      <goal ref="gShowProgress"/>
    </trigger>
  </plan>
  :
</plan>
</agent>

```

Gambar 6: Bagian kode program agen aPerformance-sProgress.agent.xml

lan data kemajuan dan perkembangan setiap kegiatan proyek, dan menyimpan sebagai dokumen status waktu pengerjaan kegiatan. Data perkembangan ini dibangkitkan secara simulasi oleh modul data generator terpisah namun dapat akses dan diatur oleh pengguna untuk menguji terhadap hasil komputasi yang dihasilkan oleh kolaborasi agen-agen dari sistem. Agen-pemonitor, bekerja sebagai agen untuk membandingkan informasi dari rencana proyek yang telah ditetapkan dengan informasi perkembangan pelaksanaan setiap kegiatan dari setiap proyek.

Agen-pengendali, bekerja sebagai pemandu awal terhadap analisis dan penentuan beberapa kemungkinan rencana tindakan terhadap pengendalian pelaksanaan proyek-proyek, dan menyimpan hasilnya sebagai dokumen analisis pola kencedrungan pelaksanaan proyek. Agen-analisis kinerja, bekerja sebagai penyiapan dan penyedia komputasi dari nilai ukuran pencapaian kinerja setiap proyek dalam berbagai bentuk laporan. Kemudian memberi alternatif pertimbangan-pertimbangan dengan melakukan rating dari setiap hasil dan kemajuan pelaksanaan proyek-proyek perusahaan.

Secara menyeluruh setiap agen-agen tersebut dikembangkan memiliki kemampuan berkolaborasi. Untuk mencapai suatu keadaan kolaborasi tertentu, setiap agen dikembangkan memiliki kemampuan berkomunikasi secara semantik dengan memanfaatkan sintaks FIPA ontologi. Selanjutnya, semua agen memiliki model atau aturan-aturan penalaran sendiri-sendiri dengan atau terhadap berbagai perkembangan pengetahuan yang diperoleh dan telah dimiliki setiap saat. Dan secara khusus didalam agen analisis kinerja didesain aturan penalaran khusus agar mampu

```

public void body() {
  EnterpriseProject EP = new EnterpriseProject();
  MonitoringJScriptGantt mjs = new MonitoringJScriptGantt();
  try {
    BufferedReader in = new BufferedReader(new
      InputStreamReader(client.getInputStream()));
    String request = in.readLine();
    int slash = request.indexOf("/");
    int space = request.indexOf(" ", slash);
    String vName = request.substring(slash + 1, space);
    PrintStream out = new
      PrintStream(client.getOutputStream());
    out.print("HTTP/1.0 200 OK\r\n");
    out.print("Content-type: text/html\r\n");
    out.println("\r\n");
    out.print(mjs.GanttHeader() +
      EP.getprogressDetail(vName));
    out.flush();
    client.close();
  } catch (IOException e) {
    throw new RuntimeException(e.getMessage());
  }
}

```

Gambar 7: Bagian kode program plan pShowProject-Progress.java

melakukan analisis kinerja dari perkembangan pelaksanaan proyek.

Sebagai contoh, ketika setiap saat dokumen analisis pola perkembangan proyek diperbaharui (updated) oleh agen pengendali, maka agen analisis kinerja akan melakukan perhitungan kembali nilai kinerja setiap proyek berdasarkan ukuran perbedaan tingkat perkembangan perubahan dengan informasi status tahapan sebelumnya dan membandingkan dengan nilai target referensi yang telah ditetapkan sebelum proyek dilaksanakan.

HASIL DAN PEMBAHASAN

Prototip model multiagen untuk komputasi kinerja proyek perusahaan ini pada dasarnya telah dibangun dengan menggunakan bahasa pemrograman java dan Agent Jadex versi 0.96, yang bersifat open source dan bebas digunakan untuk tujuan penelitian. Beberapa komponen package library java yang juga bersifat open source digunakan untuk pemrosesan data seperti ChartDirector dan jsGantt untuk pembuatan grafik atau chart. Antarmuka pada sisi pengguna dibangun dengan bahasa pemrogram Javascript versi prototype window (juga bersifat open source). Tampilan antarmuka dalam bentuk window dapat lebih memudahkan pengguna untuk mengatur dan melihat beberapa informasi proyek perusahaan secara bersamaan, tanpa perlu berpindah antar halaman web. Sedangkan untuk menangani komunikasi client-server digunakan perangkat lunak Glassfish dan Tomcat server sebagai web server.

Sebagai contoh, pada Gambar 6 diperlihatkan kode program dari file ADF (agent definition file, bahasa Jadex) dari agen aPerformancesProgress. Agen ini memiliki lima buah achieve goal, yaitu gShowProgress, gShowProjectPerformance, gDisplaySGraph, gDetailGanttProjects, gDis-



Gambar 8: Tampilan hasil kinerja proyek-proyek dalam grafik barchar dan kurva-S

play ReportStaff. Sedangkan bagian kode program java dari plan pShowProjectProgress diperlihatkan pada Gambar 7.

Pada Gambar 8 ditunjukkan tampilan dalam halaman web sebuah hasil pengujian dari prototip yang telah dibangun. Hasil ini untuk menunjukkan kemampuan dari agen dalam berkoordinasi untuk menyediakan informasi kinerja kemajuan dari seluruh proyek yang sedang dilaksanakan oleh perusahaan. Pada window Project Performances terlihat nilai kinerja yang dicapai oleh setiap proyek dalam bentuk bar chart dan tabel. Sebagai contoh Project Company High Internet (dengan nilai CPI 0.77 dan SPI 0.67 yang berarti berjalan normal), Project School Speedy I (dengan nilai CPI 1.03 dan SPI 0.96 yang berarti berkinerja cukup baik), dan Project Banana 2.0 (dengan nilai CPI 0.43 dan SPI 0.58 yang berarti berkinerja tidak bagus). Sedangkan pada window Graphics S terlihat nilai kinerja kumulatif dalam periode waktu bulanan untuk Project School Speedy I dalam bentuk kurva S.

SIMPULAN

Penelitian ini telah berhasil membangun sebuah prototip aplikasi cerdas berbasis model sistem multiagen dan dapat dijalankan didalam lingkungan web. Terdapat lima agen cerdas yang dapat bekerjasama telah dikembangkan dengan tujuan agar mampu melakukan proses komputasi kinerja kemajuan dari pelaksanaan proyek-proyek perusahaan. Seperti agen-ekstraktor, agen-pelacak-status, agen-monitor, agen-pengendali, dan agen-analisis-kinerja. Berdasarkan pada beberapa data simulasi dan hasil pengujian terhadap prototip yang dibangun menunjukan bahwa sistem multiagen telah mampu melakukan komputasi numeris untuk nilai-nilai kinerja kemajuan dari pelaksanaan proyek-proyek perusahaan. Informasi kinerja yang diperoleh ditampilkan dalam bentuk grafik barchart dan grafik S.

DAFTAR PUSTAKA

- [1] Wu, D.J., Sun, Y. and Zhong, F.: *Organizational Agent Systems for Intelligent Enterprise Modelling*. Electronic Markets **10**(4) (2000) 272–281

- [2] Bergamaschi, S., Gelati, G., Guerra, F. and Vincini, M.: *An Intelligent Data Integration Approach for Collaborative Project Management in Virtual Enterprises*. Springer Science + Business Media, Inc. Manufactured in The Netherlands (2005)
- [3] Mohammadian, M.: *Advances in Intelligent Systems: Theory and Applications*. Frontiers in Artificial Intelligence and Applications **59** (2000)
- [4] Yan, Y., Kuphal, T. and Bode, J.: *Application of Multiagent Systems in Project Management*. Int. J. Production Economics **68** (2000) 185–197
- [5] Jarvis, D., Jarvis, J., McFarlane, D., Lucas, A. and Ronnquist, R.: *Implementing a Multi-Agent Systems Approach to Collaborative Autonomous Manufacturing Operations*. IEEE Aerospace Conference (2001)
- [6] Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P. and Odgers, B.: *Autonomous Agents for Business Process Management*. Int. Journal of Applied Artificial Intelligence **14**(2) (2000) 145–189
- [7] Kaminka, G., Pynadath, D. and Tambe, M.: *Autonomous Agents for Business Process Management*. In: Proc. of Autonomous Agents'01, Montreal, Canada. (2001) 308–315
- [8] Wooldridge, M.: *An Introduction to Multi-Agent Systems*. John Wiley and Sons (2002)
- [9] Goethert, W.: *Developing enterprise-wide measures for tracking performance of acquisition organizations*. Technical report, Software engineering Institute, Carnegie Mellon University, Pittsburgh (2002)
- [10] Marchewka, J.T.: *Information Technology Projects Management: Providing Measurable Organization Value*. John Wiley and Sons, Inc, New York (2003)
- [11] Danilovic, M. and Sandkull, B.: *Managing Complexity and Uncertainty in a Multiproject Environment*. In: Proceeding of The Third Dependence Structure Matrix, International Workshop, MIT, Massachusetts, Boston. (2001)
- [12] Sycara, K.: *Multiagent systems*. Magazine (1998)
- [13] El-Khouly, M.M.M.: *An Agent-based System for Teaching Programming Languages and Reuse of Software Components*. PhD thesis, Graduate School of Science and Engineering, Saitama University, Japan (2000)
- [14] Chavez, A., Alexandros M. and Pattie M.: *Challenger: A Multi-agent System for Distributed Resources Allocation*. In: Agents '97 Conference Proc., ACM. (1997)
- [15] Pitt, J., Matthew A. and Jim C.: *Autonomous Agents in Inter-Organizations Project Management*. Communications of ACM on Intelligent Agents **37**(7) (1995)

- [16] Qomariah, N.N: Penerapan sistem agen cerdas untuk pengendalian pengeluaran biaya proyek. Master's thesis, S1 Ilmu Komputer, Fakultas MIPA UGM (2006)
- [17] Matsasinis, N.F, and Delias, P.: Implementing an agent-based decision support system for task allocation: a multi-criteria approach. Technical report, Decision Support Systems Laboratory, Technical University of Crete 73100, Chania, Greece (2003)
- [18] [24] Zha, X.F., Lim, S.Y.E, Wen and Lu, W.F.,: *A Knowledge Intensive Multi-Agent System for Cooperative/Collaborative Assembly Modeling and Process Planning*. Transactions of the SDPS **7**(1) (2003) 99–122
- [19] Gerber, A. and M. Klusch: *Agent-Based Integrated Services for Timber Production and Sales, German Research Center for Artificial Intelligence*. I2EEE Distributed Online (September, 2001)
- [20] Russell, S. J. and Norvig, P., : *Artificial Intelligence: A Modern Approach*. Prentice Hall (2003)
- [21] Azhari, Sri Hartati: *Model Komputasi Kecerdasan Berbasis Sistem Agen dan Sistem Multiagen*. In: Prosiding Seminar Nasional Manajemen Teknologi III, Peran Manajemen Teknologi Dalam Mempertahankan Sustainability Bisnis Dalam Era Pasca Kenaikan BBM. (2006) C61–C71
- [22] Wysocki, R.K. and McGary,R: *Effective Project Management*. John Wiley and Sons (2003)
- [23] Stratton, R.W.: *Applying Earned Schedule Analysis to EVM Data for Estimating Completion Date*. AACE International Transactions **4** (2007) 1–3
- [24] Lukas, J.A: *Earned Value Analysis Why it Doesn't Work*. AACE International Transactions **1** (2008) 308–315