

Pemodelan Kebutuhan Sistem Informasi Menggunakan TROPOS

Falahah¹⁾, Iping S. Suwardi²⁾, Kridanto Surendro³⁾

¹⁾Program Studi Teknik Informatika Universitas Widyatama

^{1,2,3)}Sekolah Teknik Elektro dan Informatika ITB

e-mail: falahah@widyatama.ac.id

Abstrak

Analisis kebutuhan sistem merupakan fase penting dalam siklus hidup pengembangan sistem. Kegagalan dalam mengidentifikasi kebutuhan dapat menyebabkan sistem atau software yang dihasilkan tidak dapat dimanfaatkan oleh pengguna. Saat ini terdapat banyak metode dan pendekatan dalam memodelkan kebutuhan sistem, misalnya yang berorientasi object (OOAD), RUP, Flow oriented, data oriented dan lain-lain. Pada umumnya model-model tersebut diterapkan pada proses bisnis yang pada kenyataannya seringkali melibatkan sistem pengaturan peran yang rumit. Salah satu pendekatan dalam pemodelan ini adalah pemodelan berbasis agent (Agent oriented software engineering, AOSE). AOSE sendiri terus berkembang dan sudah banyak pendekatan, metode maupun alat bantu yang sudah dipublikasikan oleh beberapa peneliti sebelumnya. Salah satunya adalah TROPOS. TROPOS memodelkan siklus hidup pengembangan software dalam 5 fase yaitu early requirement, late requirement, architectural design, detail design dan implementation. Perbedaan metode TROPOS dengan metode SDLC lainnya terletak pada fase analisis yang dibagi menjadi dua tahap yaitu early dan late requirement. Pada penelitian ini, metode TROPOS akan digunakan untuk memodelkan analisis kebutuhan sistem informasi seleksi pegawai. Pemodelan hanya dibatasi pada dua fase awal yaitu early requirement dan late requirement. Berdasarkan model early requirement diperoleh goal utama sistem, aktor dan resource yang dibutuhkan. Pemodelan late requirement dapat mengidentifikasi softgoal sebagai reason diperlukannya suatu fitur pada sistem. Hasil akhir pemodelan berupa daftar aktor, fitur-fitur utama sistem dan identifikasi kebutuhan elemen data.

Kata kunci: TROPOS, agent, model, kebutuhan sistem, aktor.

1. Pendahuluan

Identifikasi kebutuhan merupakan tahapan penting pada pengembangan sistem informasi. Kekeliruan dalam menentukan kebutuhan sistem dapat menghasilkan sistem yang tidak berguna. Teknik identifikasi kebutuhan sendiri sudah banyak dikembangkan dan menggunakan beberapa pendekatan misalnya *object-oriented*, *flow oriented*, *data oriented* dan lain-lain. Masing-masing teknik tersebut dapat diterapkan pada kondisi dan karakteristik sistem/software yang kelak akan dikembangkan. Sebagai contoh, teknik *object-oriented* cocok diterapkan pada sistem berbasis object, demikian juga *flow oriented* banyak digunakan pada paradigma pengembangan *software* yang bersifat prosedural.

Teknik pengembangan *software* saat ini juga sudah mengadopsi pendekatan berbasis *agent*. Pada pengembangan *software* berbasis *agent*, diperlukan teknik khusus untuk mengidentifikasi kebutuhan yang didasari oleh peranan masing-masing *agent* pada sistem/*software*. *Tropos* merupakan salah satu model identifikasi kebutuhan *software* berbasis *agent* yang secara keseluruhan mencakup siklus hidup pengembangan *software*, hanya pada fase identifikasi kebutuhan, dibagi menjadi 2 tahap, yaitu *early requirement* dan *late requirement*. Adanya dua tahap ini dapat membantu penyederhanaan model dan memberikan kesempatan mendetilkkan kebutuhan pada fase selanjutnya. Pada penelitian ini dicoba diterapkan model *Tropos* untuk mengidentifikasikan kebutuhan sistem informasi.

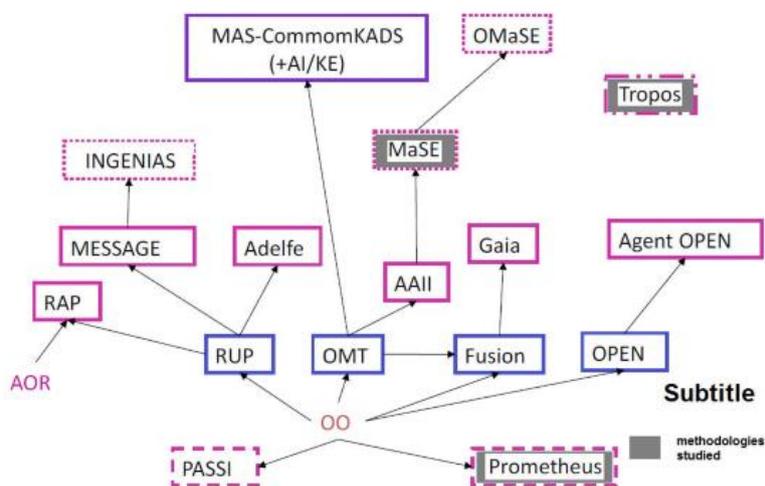
2. Model Tropos

2.1. Pemodelan Software Berbasis Agent

Pada umumnya, pemodelan software memerlukan metodologi yang meliputi dua hal yaitu abstraksi untuk menggambarkan produk dan tools untuk membantu proses abstraksi itu sendiri. Sebagai contoh metodologi misalnya pada pendekatan berorientasi object, dikenal pendekatan abstraksi menggunakan pendefinisian konsep *object*, *class*, *inheritance* dan *services*. Sedangkan metodologi pada

pengembangan *software* berorientasi *object* (OOSE, *object oriented software engineering*) misalnya OOAD, RUP, OPEN dan lain-lain, yang umumnya berpusat pada abstraksi berorientasi *object*[1]. *Tools* atau teknik pemodelan yang umum digunakan pada OOSE misalnya UML (standar), ERD, *class*, *finite state automata*, Bahasa visual dan lain-lain.

Seiring dengan perkembangan teknologi, maka pendekatan dan metodologi dalam pengembangan *software* juga berkembang. Salah satunya dengan munculnya komputasi berbasis *agent*. Komputasi berbasis *agent* sendiri merupakan satu bidang penelitian yang berfokus pada pengembangan *software* cerdas berbasis konsep “agents”[2]. Berkembangnya teknologi komputasi berbasis *agent* ini memunculkan paradigma baru dalam pengembangan *software*, yang dikenal sebagai *Agent Oriented Software Engineering* (AOSE). Munculnya paradigma ini dikarenakan kompleksitas dan distribusi *agent* pada sistem multi *agent* (*multi agent system*, MAS) tidak dapat dimodelkan dengan teknik pengembangan *software* yang sudah ada [3]. Terkait dengan bidang OASE, hingga saat ini sudah banyak metoda, model dan notasi yang diusulkan, baik yang berangkat dari pendekatan sebelumnya (misalnya, pendekatan berorientasi *object*) maupun pendekatan baru. Brandao pada [1] menggambarkan keterkaitan beberapa pendekatan pada AOSE seperti pada gambar 1. Pada gambar terlihat bahwa Tropos dikembangkan secara terpisah, dan tidak didasari oleh pendekatan lain yang sudah ada sebelumnya.



Gambar 1. Beberapa pendekatan pada *Agent Oriented Software Engineering* [1]

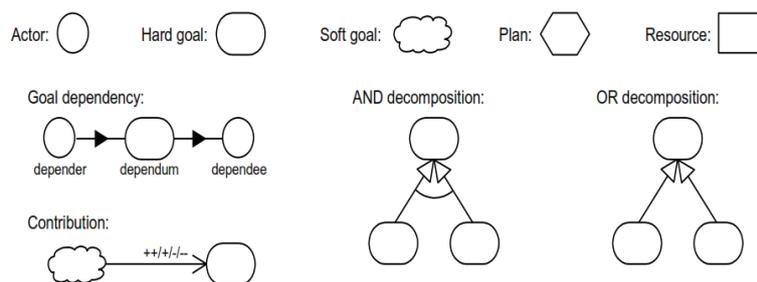
2.2. Model Tropos

Tropos, yang pertama kali dimunculkan oleh Bresciani [4], merupakan metodologi untuk pengembangan *software* yang dibagi menjadi 5 fase mulai dari *early requirement*, *late requirement*, *architectural design*, *detailed design* dan *implementation*. Perbedaan *Tropos* dengan metode pengembangan *software* lainnya terletak pada fase awal, yang membagi 2 fase *requirement* yaitu *early requirement* dan *late requirement*. Pada fase *early requirement*, analist akan mengidentifikasi domain *stakeholder* dan memodelkannya sebagai aktor sosial yang saling terkait satu sama lain berdasarkan goal yang akan dicapai, rencana yang akan dikerjakan, dan sumberdaya yang diperlukan. Sasarannya agar dapat mendefinisikan *why*, selain dari *what* dan *who*, atas kebutuhan fungsionalitas sistem. Pada fase *late requirement*, model sebelumnya kemudian dikembangkan dengan menambahkan aktor baru yang mewakili sistem, beserta aktor-aktor lain yang terkait dengan lingkungan. Secara keseluruhan, dua fase ini diharapkan dapat mewakili kebutuhan fungsional dan non fungsional dari sistem yang akan dibangun. Fase-fase lainnya seperti *architectural design*, *detailed design* dan *implementation* secara konseptual hampir serupa dengan proses SDLC (*software development life cycle*) pada umumnya.

Pemodelan sistem menggunakan Tropos didasari oleh metamodel konseptual atas elemen-elemen berikut [4]:

1. Aktor: memodelkan entitas yang memiliki tujuan strategis dan terlibat aktif pada sistem. Aktor dapat berupa *agent* fisik, sosial, atau *software* atau *role* (peranan) atau posisi. Pada *Tropos*, *role* didefinisikan sebagai karakteristik abstrak dari perilaku aktor sosial, dan posisi menyatakan sekumpulan *role* yang biasanya dimainkan oleh satu *agent*. *Agent* dapat menempati satu posisi, dan satu posisi dapat meliputi sebuah *role*.

2. *Goal*: menyatakan ketertarikan strategis aktor. *Goal* dibedakan menjadi *hard goal* dan *soft goal*. *Soft goal* adalah goal yang tidak memiliki definisi atau kriteria yang jelas atas tercapai/tidaknya. *Soft goal*, pada pengembangan software, dapat diibaratkan dengan kebutuhan non-fungsional.
3. *Plan*, menyatakan bagaimana melakukan sesuatu. Melaksanakan *plan* dapat berarti berusaha memenuhi suatu *goal*.
4. *Resource*, merupakan entitas fisik atau informasi
5. *Dependency*, antar 2 aktor, menyatakan bahwa satu aktor, untuk mencapai *goal*, melaksanakan suatu rencana, membutuhkan/menghasilkan *resource*, bergantung pada aktor lain. Aktor yang bergantung pada aktor yang lain disebut sebagai *depender*, sedangkan aktor yang dijadikan tempat bergantung disebut sebagai *dependee*. *Object* di sekitar pusat saling kebergantungan ini disebut sebagai *dependum*.
6. *Capability*, menyatakan kemampuan aktor untuk mendefinisikan, memilih dan mengeksekusi rencana untuk memenuhi tujuan, berdasarkan kondisi dan berdasarkan kejadian tertentu
7. *Belief*, menyatakan pengetahuan aktor pada domain tersebut.



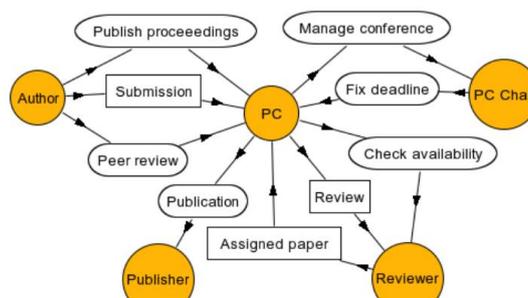
Gambar 2. Simbol-simbol pada Model Tropos [5]

Pada perkembangannya, *Tropos* sudah banyak digunakan untuk menganalisis requirement pada proses bisnis, kebutuhan dan aspek hukum dari pengembangan software pada organisasi, dan aspek keamanan. Beberapa ekstensi pada *Tropos* umumnya berupa penambahan notasi khusus untuk merepresentasikan beberapa pengembangan dari konsep dasar *Tropos* [2].

2.3. Analisis Kebutuhan dengan Tropos

Model *Tropos* menggunakan dua fase dalam identifikasi kebutuhan yaitu *early requirements* dan *late requirements*. *Early requirement* menjelaskan kondisi domain saat ini dan *late requirement* menyatakan domain yang akan disiapkan berikutnya (*system-to-be*).

Mengidentifikasi *Early Requirement* dapat dimulai dari menjawab pertanyaan berikut: siapakah *stakeholder* pada domain tersebut? Apa *goal* mereka dan bagaimana mereka terkait satu sama alian? Apakah strategi yang mengaitkan antar aktor untuk mencapai tujuannya. Pemodelan *Early requirement* dilakukan dengan cara menggambarkan diagram yang terdiri atas *actor*, *goal*, *resource* yang dibutuhkan untuk mencapai *goal* tersebut dan relasi antar berbagai komponen [4]. Gambar 3 menampilkan contoh pemodelan *early requirement* dengan lingkaran menyatakan aktor, elips menyatakan *goal* dan kotak menyatakan *resource*.



Gambar 3. Contoh Pemodelan Early Requirement [6]

Tahapan pemodelan selanjutnya yaitu dengan cara mendekomposisi sebuah goal menjadi beberapa sub goal dan mencari beberapa alternative untuk mencapai goal [4]. Alternatif dapat dinyatakan dengan dekomposisi OR dan dicirikan oleh kontribusi yang lebih dari satu. Pada fase ini, kebutuhan non

fungsional dapat dinyatakan sebagai soft-goal. Pada fase ini, terdapat kemungkinan untuk membandingkan beberapa alternative untuk mencari yang terbaik. Teknik yang dilakukan untuk mendetilkkan kebutuhan yaitu dengan cara merinci sub goal dan relasi yang lebih detil antar dua actor misalnya. Masalah lain yang harus ditetapkan adalah sudut pandang dalam menganalisa goal, misalnya berdasarkan kepentingan atau tanggung jawab tertentu. Goal kemudian didekomposisi menjadi beberapa sub goal. Soft goal dapat didefinisikan di dalam diagram actor-goal, dengan relasi kontribusi dari/ke arah goal. Goals juga dapat didekomposisi menggunakan AND untuk menyatakan goal yang saling terkait dan harus dilaksanakan (bukan merupakan pilihan). Hasil akhir fase *Early Requirement* adalah sekumpulan diagram actor dan goal.

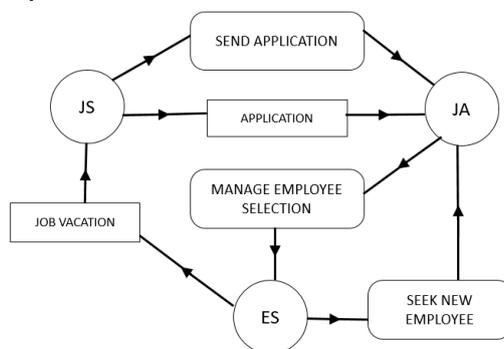
Fase Late Requirement ditujukan untuk mengidentifikasi perubahan pada domain sebagai akibat adanya sistem baru dan property sistem yang actual. Fase ini dimulai dengan memperkenalkan model domain actor baru yang dinyatakan oleh sistem. Actor baru ini dapat berupa peranan (role) yang diambil alih oleh sistem. Pertanyaan yang dapat dijadikan pedoman awal bagi fase Late Requirement adalah sebagai berikut: Goal apa saja yang dapat ditugaskan pada sistem baru dan kebergantungan apa yang dapat dipindahkan dari actor domain kepada sistem.

3. Analisis Kebutuhan Sistem Informasi

Untuk lebih jelasnya, Model Tropos ini akan digunakan untuk memodelkan analisis kebutuhan sebuah sistem informasi yaitu sistem informasi seleksi pegawai. Sistem ini dibangun untuk membantu mengkoordinasikan beberapa pihak yang terlibat dalam proses seleksi pegawai (*stake holder*) yaitu Pencari kerja (*Job Seeker / JS*), Agen Pengelola Lowongan Kerja (*Job Agent, JA*), dan Perusahaan yang membutuhkan tenaga kerja (*Employee Seeker, ES*). Secara umum, Agen pengelola lowongan kerja akan memfasilitasi proses seleksi tenaga kerja dan menerima banyak permintaan tenaga kerja dari berbagai perusahaan. Agen ini akan mengumumkan lowongan kerja, membantu menyeleksi karyawan dan memberikan rekomendasi kepada perusahaan atas hasil seleksi karyawan.

3.1. Early Requirement

Berdasarkan deskripsi di atas, dapat disimpulkan bahwa aktor pada sistem ini adalah Job Seeker (JS), Job Agent (JA) dan Employee Seeker (ES). Goal yang diharapkan oleh para stakeholder adalah dapat mengirimkan aplikasi/lamaran (*Send application*), mengelola seleksi karyawan (*manage employee selection*), dan mencari karyawan baru (*seek new employee*). Gambar 4 menampilkan model analisis kebutuhan untuk fase Early Requirement

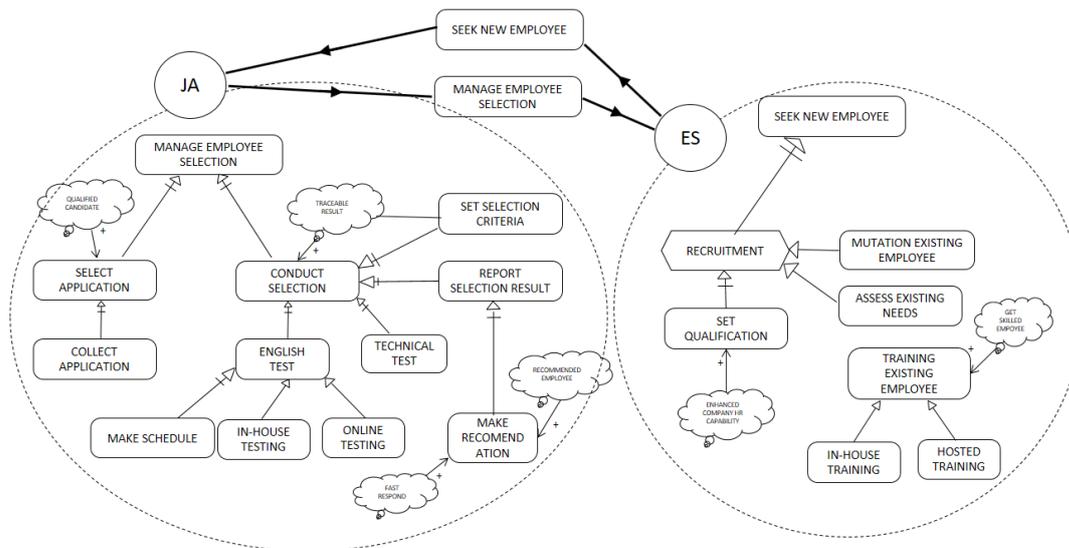


Gambar 4. Model Early Requirement

3.2. Late Requirement

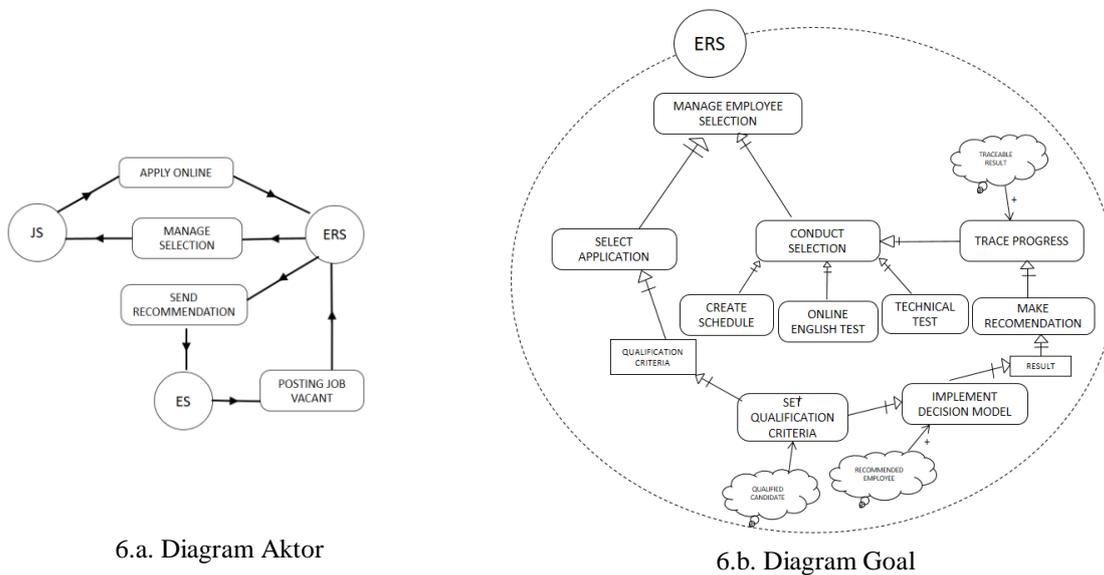
Selanjutnya akan dilakukan identifikasi rinci goal-goal di atas dengan cara melakukan dekomposisi atas goal-goal utama menjadi beberapa sub goal sesuai dengan fungsi masing-masing aktor. Sebagai contoh, goal “manage employee selection” dapat diturunkan menjadi beberapa sub goal seperti *conduct selection*, *select application*, *collect application*, *set selection criteria* dan lain-lain. Goal tersebut juga dapat dihubungkan dari satu stakeholder kepada stakeholder lain yang relevan. Gambar 5 menampilkan model goal untuk 2 aktor di atas.

Berdasarkan hasil analisis di atas, selanjutnya dapat dibuatkan model kedua yaitu model aktor dan model goal. Pada late requirement ini telah dapat diidentifikasi bahwa pada sistem yang akan dibuat yaitu ERS (*Employee Recruiting System*) yang akan menggantikan peranan JA untuk memfasilitasi pengumuman lowongan kerja, proses lamaran kerja dan memberikan rekomendasi kepada perusahaan pencari tenaga kerja (*Employee Seeker / ES*). Model aktor untuk Late Requirement dapat dilihat pada gambar 6.a.



Gambar 5. Model Goal Aktor JA (Job Agent) dan ES (Employee Seeker)

Pada model goal, “aktor” ERS atau Employee Recruiting System yang akan menggantikan peranan Job Agent, dilengkapi dengan model pengambilan keputusan agar dapat memberikan rekomendasi yang lebih representative kepada perusahaan pencari tenaga kerja (Employee Seeker). Sistem ini juga memfasilitasi penelusuran progress lamaran baik bagi pencari kerja (Job Seeker) maupun Employee Seeker. Beberapa hal yang dapat dibantu oleh sistem antara lain penyusunan jadwal seleksi dan test Bahasa Inggris secara online. Hal ini diharapkan dapat mempercepat proses seleksi sehingga pelamar pekerjaan cukup mengikuti ujian teknis terkait posisi yang dilamar. Gambar 6.b. menampilkan Goal Diagram untuk aktor ERS.



6.a. Diagram Aktor

6.b. Diagram Goal

Gambar 6. Model Late Requirement

3.3. Kebutuhan Sistem dan Kebutuhan Data

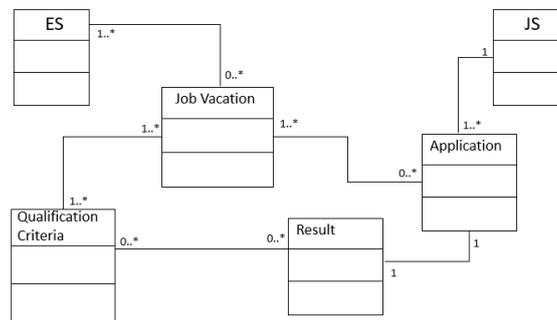
Berdasarkan model di atas, dapat diidentifikasi kebutuhan sistem informasi berdasarkan aktor-aktor atau stakeholder yang menggunakan sistem seperti pada tabel 1

Tabel 1. Kebutuhan Sistem ERS (Employee Recruiting System)

<i>Actor</i>	<i>Goal</i>	<i>Sub goal [system features]</i>
ERS (Employee Recruiting System)	manage employee selection	select applicant set qualification criteria

		conduct selection implement decision model trace progress make recommendation
Employee Seeker (ES)	Seek new employee	set qualification mutation existing employee aseses existing needs training existing employee
Job Seeker	Apply job vacation	send application attend the selection process tracking progress get the selection result

Berdasarkan resource yang digunakan untuk mencapai goal-goal di atas, dapat diidentifikasi model class diagram untuk ERS seperti pada gambar 8. Pada penelitian sebelumnya [2,3,4], teknik pemodelan TROPOS belum dikembangkan hingga mengidentifikasi class diagram.



Gambar 8. Class Diagram ERS

4. Kesimpulan

Berdasarkan uraian sebelumnya dapat disimpulkan bahwa pendekatan Tropos, meskipun berangkat dari pendekatan pemodelan untuk sistem berbasis agent, ternyata dapat diterapkan pada pemodelan sistem informasi secara umum, dengan cara mengidentifikasi aktor-aktor pada sistem serta goal dari masing-masing aktor. Hasil penelitian menunjukkan bahwa melalui identifikasi goal, dapat diidentifikasi fitur-fitur sistem yang akan dikembangkan (melalui identifikasi sub goal) dan class-class pada sistem (melalui identifikasi resource system). Kesulitan dalam penggunaan pendekatan Tropos adalah membedakan antara softgoal dan goal, sehingga untuk setiap goal yang berhasil diidentifikasi, perlu ditetapkan kriteria pengukuran pencapaian (Key performance indicator, KPI), agar memudahkan dalam menyusun softgoal dan goal utama. Jika suatu goal tidak memiliki KPI yang jelas, maka dapat dikategorikan sebagai softgoal.

Daftar Pustaka

- [1] Cunha, Raffael & Adamatti, Diana & Billa, Cléo. (2015). Agent oriented software engineering: a comparative study between methodologies that support the development of multi-agent systems. 48-52. 10.1145/2857218.2857226.
- [2] Sampathkumar, TT., Venkateswaran, V., Vishnu, P., 2014, Agent Based Distribution Tutoring System Modeling Using Extended Tropos Approach, IOSR Journal of Computer Engineering (IOSR-JCE) Vol. 16, issue 1, Feb 2014, pp35-43
- [3] Henderson-Sellers, B., 2013, Towards the Consolidation of a Diagramming Suite for Agent-Oriented Modelling Language, ISRN Software Engineering Vol 2013, article id. 803638
- [4] Bresciani, P., Perini, A., 2004, Tropos: An Agent-Oriented Software Development Methodology, Autonomus Agents and Multi-Agent Systems, Vol 8., pp 203-236, Kluwer Academic Publishers.
- [5] Bertolini, D., Mylopoulos, J., Delpero, L., Tomasi, B., 2006, A Tropos Model-Driven Development Environment, Proceeding of 18Th Conference on Advanced Information System Engineering (CAiSE 06), Luxembourg, June 5-9, 2006
- [6] Moradini, M., Nguyen, D.C., Perini, A., Siena, A., Susi, A., 2007, Tools-Supported Development with Tropos: The Conference Management System Case Study, International Workshop on Agent-Oriented Software Engineering, Lecturer Notes in Cumputer Science (LCNS, vol.4951)