

Rekayasa Kebutuhan dengan Metode Pemodelan Berbasis Linguistik dan Ontologi pada Sistem Penilaian Prestasi Kerja Pegawai Dinas Kominfo Kota Malang

Nurwida Mariatul Sadila¹, Fajar Pradana², Bayu Priyambadha³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email : nurwidamsadila@gmail.com¹, fajar.prd@gmail.com², bayu_priyambadha@ub.ac.id

Abstrak

Rekayasa kebutuhan perangkat lunak merupakan hal yang penting dalam proses rekayasa perangkat lunak. Salah satu faktor utama kegagalan dalam rekayasa perangkat lunak adalah pada tahap rekayasa kebutuhan tidak dilakukan dengan baik dan benar. Pemberian struktur dan semantik yang jelas akan menjadikan spesifikasi kebutuhan mudah dipahami dan divalidasi. Metode linguistik dan ontologi dapat digunakan sebagai bahan untuk memodelkan spesifikasi kebutuhan dan validasinya. Studi kasus yang digunakan pada penelitian ini adalah Sistem Penilaian Prestasi Kerja Dinas Kominfo Kota Malang. Spesifikasi kebutuhan yang didapat pada proses pembangunan sistem diperbaharui dan dimodelkan ke dalam bentuk semantik. Metode semantik yang digunakan adalah DITA (Darwin Information Typing Architecture) yang berfungsi untuk mendokumentasikan kebutuhan dalam bentuk skenario dengan tambahan anotasi ontologi. Tambahan anotasi ontologi berfungsi sebagai acuan dari DITA ke ontologi maupun sebaliknya. Hal ini untuk mempermudah pemahaman kebutuhan ketika terjadi perawatan perangkat lunak. Validasi dan verifikasi kebutuhan menggunakan metode ontologi yaitu dengan melakukan *query* terhadap kebutuhan tersebut. Hasil *query* tersebut menunjukkan status *completeness*, *consistency*, dan *correctness* terhadap kebutuhan. Dengan melakukan validasi dan verifikasi kebutuhan menggunakan metode ini, kebutuhan yang memiliki kesalahan sebelumnya dapat segera terdeteksi dengan mudah dan memiliki tingkat *completeness*, *consistency*, dan *correctness* sebesar 100%.

Kata kunci : spesifikasi kebutuhan, validasi dan verifikasi kebutuhan, semantik, DITA, ontologi.

Abstract

Software requirements engineering is crucial in the software engineering process. One of the main causes of software requirements engineering failure is that the stages are not done quite well. The structuring and clearly defined semantic will make the requirement specification easy to be understood and validated. Linguistic based modeling method and ontology be used for modeling specification and its validation base. Sistem Penilaian Prestasi Kerja Dinas Kominfo Kota Malang is used as case study in this thesis. The requirement specification which is obtained in the development process is modified and modeled into semantic model. DITA (Darwin Information Typing Architecture) is used as semantic method which work for documenting the requirements in scenario format with additional ontology annotations. The additional ontology function is to be the reference from DITA to ontology or vice versa. It aims to facilitate the requirements understanding during the maintenance process. The requirements validation and verification is using the ontology method by applying the query to the requirements. The query results show the completeness, consistency, and correctness status toward the requirements. The benefits of performing the validation and verification to the requirements using this method are the requirements which have errors can be detected very soon and easily with the level of completeness, consistency, and correctness up to 100%.

Keyword : requirements specification, requirements validation and verification, semantic, DITA, ontology.

1. PENDAHULUAN

Rekayasa perangkat lunak merupakan penerapan sebuah pendekatan yang sistematis, tertib, dan terukur terhadap pengembangan, pengoperasian, dan perawatan perangkat lunak (Laplante, 2007). Untuk merekayasa perangkat lunak tersebut umumnya terdapat tahapan yaitu menganalisis kebutuhan pengguna, selanjutnya merancang desain dari hasil analisis kebutuhan, kemudian mengimplementasikan desain yang telah dibuat, dan tahap menguji perangkat lunak. Sebelum memulai tahap pengembangan perangkat lunak, kebutuhan dari pengguna harus diklarifikasi dan didetailkan menjadi spesifikasi kebutuhan. Spesifikasi kebutuhan ini memuat semua atau sebagian besar rincian layanan dari produk perangkat lunak yang akan dikembangkan dan secara umum dijelaskan dengan bahasa natural (Lautenbacher, 2010).

Salah satu faktor utama kegagalan dalam rekayasa perangkat lunak yaitu pada tahap rekayasa kebutuhan tidak dilakukan dengan benar terutama pada saat membuat spesifikasi kebutuhan (Hoffman & Lehner, 2001). Kegagalan tersebut menunjukkan bahwa urgensi dari rekayasa kebutuhan itu sendiri sangatlah tinggi pada proses rekayasa perangkat lunak. Namun untuk merekayasa kebutuhan dengan tepat itu sendiri masih terbilang sulit (Swiss, 2012).

Jika spesifikasi kebutuhan memiliki struktur dan semantik yang jelas maka kemungkinan besar kendala kegagalan proses rekayasa perangkat lunak yang disebabkan oleh proses rekayasa kebutuhan akan mudah diatasi. Oleh karena itu terdapat suatu metode yang dapat memudahkan proses rekayasa kebutuhan yaitu pemodelan berbasis linguistik dan ontologi. Dengan metode ini akan dibuat strukturisasi pada spesifikasi kebutuhan dan memberikan anotasi pada istilah yang digunakan untuk meningkatkan pemahaman dan penggunaan kembali dokumen-dokumen selama proses rekayasa perangkat lunak (Lautenbacher, 2010).

Penilaian prestasi kinerja adalah suatu proses penilaian prestasi kinerja pegawai yang dilakukan pemimpin perusahaan secara sistematis berdasarkan pekerjaan yang ditugaskan kepadanya (Dessler, 1997). Salah satu institusi yang mempunyai sistem penilaian kinerja pegawai adalah Dinas Kominfo Kota Malang. Awalnya penilaian kinerja pegawai pada dinas ini dilakukan secara manual,

kemudian dibuatlah sebuah sistem berbasis web untuk memudahkan proses penilaian. Pada pembangunannya, sistem tersebut menggunakan metode *waterfall lifecycle development system*. Namun, pada saat memasuki fase implementasi dan pengujian terdapat beberapa fitur yang belum sesuai dengan kebutuhan pengguna sehingga menyebabkan adanya pengulangan pada tahapan-tahapan analisis dan desain. Hal ini tentunya disebabkan karena pada saat proses analisis kebutuhan tidak dilakukan dengan baik dan benar.

Berdasarkan masalah tersebut, maka dibuatlah penelitian tentang rekayasa kebutuhan dengan metode pemodelan berbasis linguistik dan ontologi pada Sistem Penilaian Prestasi Kerja Pegawai Dinas Kominfo Kota Malang. Penelitian ini akan dilakukan dengan cara mengkaji ulang spesifikasi kebutuhan yang telah dibuat pada proses pembuatan perangkat lunak dengan cara strukturisasi dan memberikan anotasi pada istilah-istilah tertentu dan validasi ontologi yang akan menghasilkan spesifikasi kebutuhan yang lebih baik. Dengan demikian kemungkinan untuk menghasilkan perangkat lunak yang tidak sesuai dengan kebutuhan pengguna akan semakin berkurang.

2. LANDASAN KEPUSTAKAAN

Teknik penulisan skenario banyak digunakan oleh seorang analis sistem dalam menuliskan kebutuhan. Kualitas dari kebutuhan itu sendiri merupakan faktor penting dalam efisiensi dan keberhasilan pengembangan suatu perangkat lunak (Alspaugh, 2008). Pada penelitian sebelumnya yang ditulis oleh Florian Lautenbacher dkk. yang berjudul "*Linguistics-Based Modeling Methods and Ontologies in Requirements Engineering*" membahas tentang bagaimana cara memodelkan suatu spesifikasi kebutuhan dengan menggunakan metode linguistik dan ontologi. Menurut penelitian tersebut memodelkan suatu skenario spesifikasi kebutuhan yang berbahasa natural dipetakan ke dalam arsitektur DITA beserta kebutuhan ontologi. Namun untuk penelitian tersebut tidak menunjukkan validasi dan verifikasi terhadap kebutuhan yang telah di modelkan.

Penelitian dari Haibo Hu dkk. yang berjudul "*Semantic-based Requirements Analysis and Verification*" menggunakan metode ontologi dengan *Ontology Web Language* sebagai validasi dan verifikasi

terhadap kebutuhan. Untuk membuat kebutuhan yang lebih baik maka digabungkan metode DITA dan ontologi beserta verifikasi dan validasinya.

2.2 Spesifikasi Kebutuhan Perangkat Lunak

Rekayasa kebutuhan perangkat lunak merupakan proses pembuatan layanan-layanan yang pengguna butuhkan terhadap sistem dan batasan yang akan dioperasikan pada proses pengembangan (Sommerville, 2005). Spesifikasi kebutuhan perangkat lunak merupakan rincian layanan dari sebuah sistem. Kebutuhan ini sendiri ditentukan berdasarkan masalah yang perlu diselesaikan, batasan sistem, pemangku kepentingan, dan tujuan sistem (Rouibah, 2009).

2.3 Metode Pemodelan Linguistik Dan Ontologi

Linguistik adalah salah satu ilmu yang mempelajari tentang bahasa, sedangkan semantik adalah cabang dari linguistik yang mempelajari tentang makna dari suatu bahasa (Akmajian, 2010). Pengembangan perangkat lunak berdasarkan spesifikasi kebutuhan yang disusun oleh seorang analis sistem seringkali menyebabkan kesalahpahaman karena perbedaan latar belakang dari setiap individu. Metode pemodelan semantik memberikan strukturisasi terhadap spesifikasi kebutuhan yang telah didefinisikan dan ditambah dengan penganotasian dari ontologi.

2.3.1 DITA (Darwin Information Typing Architecture)

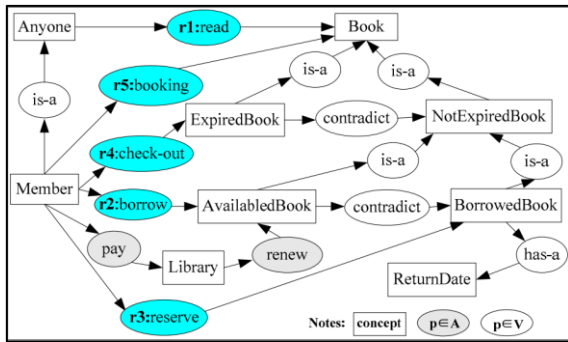
Darwin Information Typing Architecture (DITA) adalah *end-to-end* arsitektur berbasis XML untuk authoring, produksi, dan memberikan informasi teknis yang terdiri dari seperangkat prinsip-prinsip desain untuk menciptakan "Tipe Informasi" (Day, 2005). Dalam proses rekayasa perangkat lunak pemodelan DITA dapat digunakan pada proses rekayasa kebutuhan. Skenario kebutuhan yang telah dibuat dapat didokumentasikan dengan DITA agar mempermudah proses pembacaan dari setiap unsur pemangku kepentingan. Selain itu, dengan diformatkannya skenario kebutuhan dapat mempermudah proses perawatan terhadap sistem karena pengembang dapat dengan mudah mencari kata kunci dan menemukan komponen yang telah ada penjelasannya (Lautenbacher, 2010).

2.3.2 Requirement Ontology

Pendekatan ontologi pada spesifikasi kebutuhan dapat mengurangi hasil negatif seperti kebutuhan yang ambigu, spesifikasi yang tidak lengkap, tidak terdefinisi dengan jelas, dan perubahan kebutuhan yang dinamik dari proses rekayasa kebutuhan (Siddiqui, 2011). Dengan menggunakan domain ontologi, kebutuhan yang digambarkan dalam bahasa natural didekomposisi menjadi beberapa item kebutuhan yang berstruktur atomik. Kebutuhan yang dijelaskan dalam pemodelan ontologi lebih mudah dimengerti oleh pengembang maupun lingkup pengguna. Pemodelan ditambah dengan menggunakan *mapping rules* pada domain ontologi dapat menyatukan kebutuhan yang atomik dengan *semantic knowledge* (Hu, 2010).

Tahap pemodelan dengan menggunakan metode ontology yaitu memecah kebutuhan yang asli menjadi suatu set kebutuhan yang atomic dan dapat menjelaskan masalah terkecil, kemudian memetakan kebutuhan tersebut kedalam elemen dari domain ontologi, terakhir memperoleh kebutuhan dari tahap 1 dan 2 yang sudah direpresentasikan dalam bentuk model. Pada domain ontologi terdapat definisi formal yaitu $D(C, R, A^R, X)$ dengan D = domain ontologi, C = konsep pada domain ontologi, R = hubungan/relasi antar konsep, A^R = atribut dari hubungan/relasi, dan X = aksiom. Pada domain kebutuhan atomik dapat dipetakan sebagai $r(s, p, o)$ atau $p(s, o)$ dengan $s \in C$ = Subjek dari konsep kebutuhan atomic, $o \in C$ = Objek dari konsep kebutuhan atomic, dan $p \in R$ = relasi antar subjek dan objek dari kebutuhan atomik. Selain itu, terdapat dekomposisi dari A^R pada ontologi menjadi set V dan A , sehingga $p(s, o)$ dapat direpresentasikan menjadi $v(s, o)$ atau $a(s, o)$. Set V terdiri dari $\{is-a, has-a, synonym, contreadict\}$, set A terdiri relasi asosiasi. Contoh pemodelan ontologi akan ditunjukkan pada Gambar 2.1.

Parameter validasi dan verifikasi yang dapat dilakukan pada pemodelan ontologi ini yaitu *completeness*, *consistency*, dan *correctness*. Pemeriksaan dilakukan dengan menggunakan metode *inference rule* yaitu DL (*Description Logic*) (Hu, 2010). Dasar dari pemeriksaan *completeness* adalah memeriksa semua *triple* $p(s, o)$ dari sistem ontologi. Jika terdapat relasi yang tidak dipetakan antara *triple* ontologi dengan item kebutuhan, maka spesifikasi kebutuhan tersebut tidak lengkap.



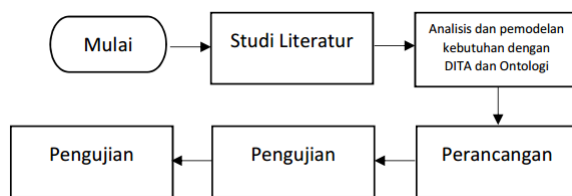
Gambar 2.1 Contoh pemodelan kebutuhan kedalam domain ontologi

Sumber : (Hu, 2010)

Dasar pemeriksaan *consistency* adalah memeriksa semua relasi p terpetakan dari kebutuhan kedalam ontologi. Jika terdapat kesamaan dua atau lebih relasi p dan subjek s , contohnya $p(s, o_1)$, $p(s, o_2)$, $p(s, o_3)$ dan seterusnya, ketika terdapat relasi *contradict* diantara o_1, o_2, o_3 , maka spesifikasi kebutuhan tersebut tidak konsisten. Dasar pemeriksaan *correctness* adalah memeriksa semua item kebutuhan pada spesifikasi kebutuhan. Jika ada item(s) kebutuhan yang tidak dapat dipetakan kedalam satu atau lebih *triple ontology*, maka spesifikasi kebutuhan tersebut tidak benar. Penerapan ontologi kebutuhan dapat dengan menggunakan *Ontology Web Language* dengan tujuan menganotasikan spesifikasi kebutuhan (Lautenbacher, 2010).

3. METODE PENELITIAN

Metodologi yang akan diterapkan pada penelitian ini dimulai dari studi literatur, analisis dan pemodelan kebutuhan dengan DITA dan Ontologi, perancangan, pengujian, serta pengambilan kesimpulan dan saran yang diilustrasikan dengan diagram alur pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Studi literatur bertujuan untuk mendukung proses penulisan penelitian dengan dasar teori agar tujuan dan pemecahan masalah dapat direalisasikan dengan baik dan benar. Teori-teori dan referensi pendukung yang diperlukan

menjadi dasar dari penelitian ini adalah kebutuhan perangkat lunak, DITA, *Requirement Ontology*, Metrik Kohesi dan Kopling.

3.2 Analisis dan Pemodelan Kebutuhan

Analisis Kebutuhan merupakan tahapan awal untuk proses rekasa perangkat lunak. Tahapan ini berisi elisitasi kebutuhan, spesifikasi kebutuhan, validasi dan verifikasi, serta manajemen kebutuhan.

3.2.1 Gambaran Umum Sistem

Sistem penilaian prestasi kerja pegawai (SPPK) adalah sebuah perangkat lunak berbasis web untuk mempermudah penilaian kinerja pegawai yang selama ini dilakukan secara manual di Dinas Kominfo Kota Malang. Dalam web ini memiliki fitur utama yaitu manajemen dari kegiatan-kegiatan yang direncanakan dan dilakukan oleh seorang pegawai.

3.2.2 Identifikasi Aktor

Tabel 3.1 menjelaskan aktor yang berinteraksi di dalam sistem beserta deskripsinya.

Tabel 3.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Dalam sistem ini pengguna dapat melakukan login untuk proses autentifikasi terhadap pengguna tersebut ketika ingin menggunakan sistem.
Admin	Dalam sistem ini admin bertugas untuk manajemen data dari pegawai dan penilai.
Pegawai	Dalam sistem ini pegawai dapat menginputkan data kegiatan dari SKP miliknya, melihat hasil penilaian dari SKP maupun PPK miliknya, mencetak hasil penilaian, dan mengubah biodata.
Penilai	Dalam sistem ini penilai dapat memberikan penilaian terhadap SKP dan PPK sesuai dengan pegawai yang dinilai olehnya, dan mengubah biodata.

3.2.3 Analisis dan Spesifikasi Kebutuhan

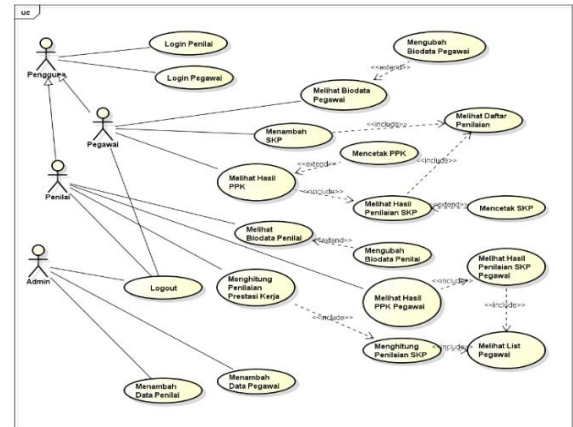
Tabel 3.2 menjelaskan potongan analisis dan spesifikasi kebutuhan yang telah diperbaharui dari pengembang sebelumnya.

Tabel 3.2 Analisis Kebutuhan

Kode	Deskripsi
SPPK-1-100	<p>P/L harus menyediakan halaman awal untuk login.</p> <p>Spesifikasi Kebutuhan :</p> <ul style="list-style-type: none"> • P/L harus membedakan halaman login pegawai dengan login penilai. Login harus menggunakan username berupa NIP dan password. (SPPK-1-101) • Untuk login pada admin, menggunakan URL yang terpisah. (SPPK-1-102)
SPPK-1-200	<p>P/L harus menyediakan halaman utama admin untuk melakukan manajemen data pegawai dan penilai.</p> <p>Spesifikasi Kebutuhan :</p> <ul style="list-style-type: none"> • P/L harus menyediakan halaman manajemen data pegawai dan penilai yang mana setiap penambahan pegawai baru harus disertakan kolom NIM dari penilai pegawai tersebut. (SPPK-1-202) • P/L harus menyediakan pilihan list daftar pegawai dan penilai yang telah dibuat berdasarkan aturan/susunan organisasi pada kantor yang bersangkutan untuk admin. (SPPK-1-203)
SPPK-1-300	<p>P/L harus menyediakan sarana untuk melakukan proses penambahan Sasaran Kerja Pegawai (SKP) oleh pegawai.</p> <p>Spesifikasi Kebutuhan :</p> <ul style="list-style-type: none"> • Untuk pegawai, P/L harus menyediakan Tambah SKP dengan nama SKP yaitu tahun pembuatan SKP. (SPPK-1-301) • Untuk pegawai, P/L harus menyediakan form untuk mengisi kegiatan yang terdiri dari nama kegiatan, kuantitas, kualitas dan waktu. (SPPK-1-302) • Untuk pegawai, P/L harus menyediakan tabel untuk menampilkan kegiatan yang sudah dibuat sesuai dengan masing-masing SKP. (SPPK-1-303) • Untuk pegawai, P/L harus menyediakan tombol submit untuk menyimpan SKP yang sudah selesai dibuat. (SPPK-1-304)

3.2.4 Use Case Diagram

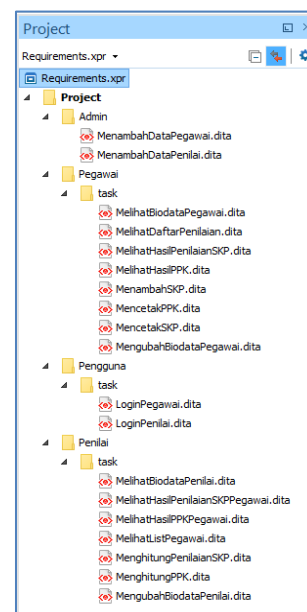
Use case diagram adalah gambaran perilaku sistem dari luar. Gambarnya berupa fungsi-fungsi yang harus dipenuhi sistem sesuai dengan aktornya. Use case diagram ditunjukkan pada Gambar 3.2.



Gambar 3.2 Use Case Diagram

3.2.5 Pemodelan DITA

Pemodelan DITA pada sistem ini didasarkan pada usecase scenario dan ditambah dengan context sesuai dengan format pemodelan DITA. Setiap skenario diberikan judul dan dikelompokkan berdasarkan aktornya. Di setiap skenario terdapat identifikasi relasi yang terhubung pada requirement ontology. Struktur dokumen DITA beserta task ditunjukkan pada Gambar 3.3.



Gambar 3.3 Struktur Dokumen DITA

Berikut adalah contoh dokumen DITA yang telah diberikan anotasi ontologi, Skenario Menambah Data Pegawai :

```

<task id="menambahDataPegawai">
  <title
  ref="Ontology_Subjek(Admin)">Menambah Data
  Pegawai</title>
  <shortdesc>Menjelaskan cara admin melakukan
  tambah data pegawai</shortdesc>
  <taskbody>
    <prereq>Admin masuk ke halaman admin
    dan memilih menu manajemen data
    pegawai</prereq>
    <context>Admin yang ingin menambahkan
    data pegawai ke dalam sistem<p>Basic Flow
    :</p></context>
    <steps>
      <step>
        <cmd>Sistem Menampilkan halaman
        manajemen data pegawai</cmd>
      </step>
      <step>
        <cmd>Admin memilih menu tambah
        data pegawai</cmd>
      </step>
      <step>
        <cmd>Sistem menampilkan halaman
        form tambah data pegawai</cmd>
      </step>
      <step>
        <cmd>Admin mengisi form tambah data
        pegawai yang salah satu kolomnya berisi NIP
        penilai dari pegawai tersebut dan menekan tombol
        "simpan"</cmd>
      </step>
      <step>
        <cmd>Sistem menyimpan data pegawai
        yang baru ke database</cmd>
      </step>
    </steps>
    <result>Data pegawai telah
    ditambahkan</result>
  </taskbody>
</task>
    
```

3.2.6 Pemodelan Ontologi

Tahapan awal dari Pemodelan Ontologi yaitu mengambil setiap item SPO(subjek, predikat, objek) dalam kebutuhan kemudian memecahnya menjadi item kebutuhan dan memetakan kedalam bentuk ontologi. Pada Tabel 3.3, subjek, predikat dan objek pada item kebutuhan diberikan kode untuk setiap definisi agar mempermudah pembentukan relasi ditahap selanjutnya.

Tahapan selanjutnya yaitu membuat relasi antar SPO(subjek, predikat, objek) dari item kebutuhan (memasukkan kedalam *inference*

Tabel 3.3 Pemetaan item kebutuhan

SPO	Kode	Definisi
Subjek	S ₁	Admin
	S ₂	Pengguna
	S ₃	Penilai
	S ₄	Pegawai
Predikat	P ₁	Login_as
	P ₂	Melihat
	P ₃	Menambah
	P ₄	Mencetak
	P ₅	Menghitung
	P ₆	Mengubah
Objek	O ₁	Biodata_pegawai
	O ₂	Biodata_penilai
	O ₃	Daftar_penilaian
	O ₄	Data_pegawai
	O ₅	Hasil_penilaian_skp
	O ₆	Hasil_penilaian_skp_pegawai
	O ₇	Hasil_ppk
	O ₈	Hasil_ppk_pegawai
	O ₉	List_pegawai
	O ₁₀	Penilaian_skp
	O ₁₁	PPK
	O ₁₂	Pegawai
	O ₁₃	Penilai
	O ₁₄	SKP
	O ₁₅	Data_penilai

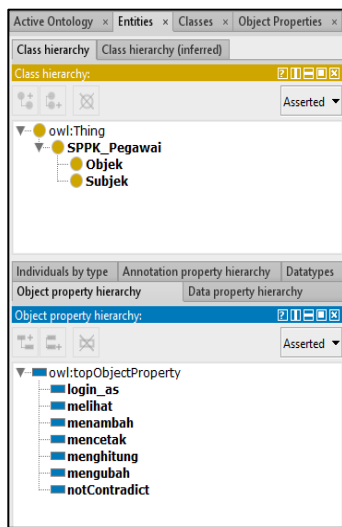
rules) sesuai dengan kalimat kebutuhan yang telah diberi kode sehingga mempermudah identifikasi. Relasi antar subjek, predikat dan objek digambarkan pada Tabel 3.4.

Tabel 3.4 Relasi antar SPO

No	Subjek	Predikat	Objek
1	S ₂	P ₁	O ₁₂
2	S ₂	P ₁	O ₁₃
3	S ₄	P ₂	O ₁
4	S ₄	P ₆	O ₁
5	S ₄	P ₂	O ₃
6	S ₄	P ₃	O ₁₄
7	S ₄	P ₂	O ₅
8	S ₄	P ₄	O ₁₄
9	S ₄	P ₂	O ₇
10	S ₄	P ₄	O ₁₁
11	S ₃	P ₂	O ₂
12	S ₃	P ₆	O ₃
13	S ₃	P ₂	O ₉
14	S ₃	P ₂	O ₆
15	S ₃	P ₂	O ₈

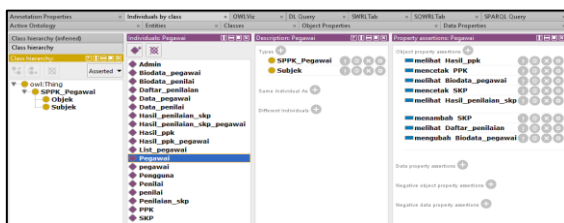
16	S ₃	P ₅	O ₁₀
17	S ₃	P ₅	O ₁₁
18	S ₁	P ₃	O ₄
19	S ₁	P ₃	O ₁₅

Dalam pemodelan ontologi ini digunakan tool Protégé dan OWL (*Ontology Web Language*) sebagai bahasa pemrogramannya. Subjek dan Objek direpresentasikan dalam *Class* dan Predikat sebagai *Object Property* pada OWL ditunjukkan pada Gambar 3.4.



Gambar 3.4 *Class dan Object Property* pada Ontologi

Setiap subjek pada item kebutuhan dimasukkan kedalam Individual yang memiliki tipe *Class* Subjek dan setiap objek pada item kebutuhan dimasukkan kedalam Individual yang memiliki tipe *Class* Objek. Subjek pada Individual tersebut akan diberikan relasi terhadap Objek sesuai dengan Tabel 3.4. Contoh Relasi Subjek Individual Pegawai terhadap beberapa Objek Individual ditunjukkan pada Gambar 3.5.



Gambar 3.5 Subjek Individual Pegawai dan relasinya Terhadap Objek Individual

3.2.5 Validasi dan Verifikasi *Requirement Ontology*

Terdapat tiga pemeriksaan terhadap kebutuhan yang dapat dilakukan dengan menggunakan ontologi. Pemeriksaan dilakukan melalui *query* terhadap kebutuhan yang telah dipetakan ke dalam ontologi. Hasil *query* tersebut dapat menunjukkan status *completeness*, *consistency* dan *correctness* terhadap kebutuhan. Tahap pertama setiap deskripsi kebutuhan diubah kedalam bentuk *triple ontology*, beberapa contoh ditunjukkan pada Tabel 3.5.

Tabel 3.5 Daftar kebutuhan yang dipetakan

Deskripsi Kebutuhan	Kebutuhan dalam <i>Triple</i>
Pengguna dapat login sebagai pegawai	login(Pengguna, Pegawai)
Pegawai dapat menambah SKP	menambah(Pegawai, SKP)
Pegawai dapat melihat hasil penilaian SKP	melihat(Pegawai, Hasil_penilaian_SKP)
Penilai dapat melihat list pegawai	melihat(Penilai, List_pegawai)
Penilai dapat menghitung PPK	menghitung(Penilai, PPK)
Admin dapat menambah data	menambah(Admin, Data_penilai)

3.2.5.1 Pemeriksaan *Completeness*

Setiap predikat dalam ontologi dijadikan sebagai acuan untuk proses *query*. Hasil *query* akan menunjukkan list kalimat kebutuhan yang ada pada ontologi kemudian disesuaikan secara manual terhadap kalimat kebutuhan. Daftar *query completeness* ditunjukkan pada Gambar 3.6.

<input checked="" type="checkbox"/>	QueryCompletenessByP1	login_as(?S, ?O) -> sqwrl:select(?S, login_as, ?O) ^ sqwrl:orderBy(?S)
<input checked="" type="checkbox"/>	QueryCompletenessByP2	melihat(?S, ?O) -> sqwrl:select(?S, melihat, ?O) ^ sqwrl:orderBy(?S)
<input checked="" type="checkbox"/>	QueryCompletenessByP3	menambah(?S, ?O) -> sqwrl:select(?S, menambah, ?O)
<input checked="" type="checkbox"/>	QueryCompletenessByP4	mencetak(?S, ?O) -> sqwrl:select(?S, mencetak, ?O)
<input checked="" type="checkbox"/>	QueryCompletenessByP5	menghitung(?S, ?O) -> sqwrl:select(?S, menghitung, ?O)
<input checked="" type="checkbox"/>	QueryCompletenessByP6	mengubah(?S, ?O) -> sqwrl:select(?S, mengubah, ?O)

Gambar 3.6 Daftar *Query Completeness*

Gambar 3.7 merupakan contoh hasil dari *query completeness* yang didasarkan oleh P2.

QueryCompletenessByP4	QueryCompletenessByP5	QueryCompletenessByP6
SQWRL Queries	OWL 2 RL	QueryCompletenessByP1
S	C1	O
Pegawai	melihat	Daftar_penilaian
Pegawai	melihat	Hasil_ppk
Pegawai	melihat	Biodata_pegawai
Pegawai	melihat	Hasil_penilaian_skp
Penilai	melihat	Biodata_penilai
Penilai	melihat	Hasil_penilaian_skp_pegawai
Penilai	melihat	List_pegawai
Penilai	melihat	Hasil_ppk_pegawai

Gambar 3.7 Hasil Query Completeness oleh P2
Completeness : $|F_m| / |O| = 19/19 = 100\%$

Dari hasil query yang diperoleh menunjukkan bahwa jumlah triple di dalam sistem ontologi yang telah terpetakan (telah diperiksa kebenarannya) menjadi item kebutuhan adalah 19 dan jumlah triple di dalam sistem ontologi adalah 19 yang kemudian dimasukkan kedalam rumus completeness sehingga mendapatkan hasil 100%.

3.2.5.2 Pemeriksaan Completeness

Setiap predikat dalam ontologi dijadikan sebagai acuan untuk proses query, kemudian hasil query akan menunjukkan list kalimat kebutuhan yang tidak berlawanan makna kemudian disesuaikan secara manual terhadap kalimat kebutuhan. Daftar query consistency ditunjukkan pada Gambar 3.8.

✓ QueryConsistencyByP1	login_as(P, ?O) ^ notContradict(?O, ?O) -> sqwrlselect(?P, login_as, ?O)
✓ QueryConsistencyByP2	melihat(P, ?O) ^ notContradict(?O, ?O) -> sqwrlselect(?P, melihat, ?O)
✓ QueryConsistencyByP3	menambah(P, ?O) ^ notContradict(?O, ?O) -> sqwrlselect(?P, menambah, ?O)
✓ QueryConsistencyByP4	mencetak(P, ?O) ^ notContradict(?O, ?O) -> sqwrlselect(?P, mencetak, ?O)
✓ QueryConsistencyByP5	menghitung(P, ?O) ^ notContradict(?O, ?O) -> sqwrlselect(?P, menghitung, ?O)
✓ QueryConsistencyByP6	mengubah(P, ?O) ^ notContradict(?O, ?O) -> sqwrlselect(?P, mengubah, ?O)

Gambar 3.8 Query Consistency

Gambar 3.9 merupakan contoh hasil dari query consistency yang didasarkan oleh P2.

QueryConsistencyByP3	QueryConsistencyByP4	QueryConsistencyByP5	QueryConsistencyByP6
SQWRL Queries	OWL 2 RL	QueryConsistencyByP1	QueryConsistencyByP2
P	C1	O	
Pegawai	melihat	Hasil_ppk	
Pegawai	melihat	Daftar_penilaian	
Penilai	melihat	Biodata_penilai	
Penilai	melihat	Hasil_penilaian_skp_pegawai	
Pegawai	melihat	Biodata_pegawai	
Penilai	melihat	List_pegawai	
Penilai	melihat	Hasil_ppk_pegawai	
Pegawai	melihat	Hasil_penilaian_skp	

Gambar 3.9 Hasil Query Consistency oleh P2
Consistency : $|\neg\text{contradict}(x,y) \wedge F_p| / |F_p| = 19/19 = 100\%$

Dari hasil query yang diperoleh menunjukkan bahwa jumlah item kebutuhan yang telah dipetakan (telah diperiksa kebenarannya) kedalam sistem ontologi dan tidak kontradiksi adalah 19 dan jumlah item kebutuhan yang telah dipetakan kedalam sistem ontologi adalah 19 yang kemudian dimasukkan kedalam rumus consistency sehingga mendapatkan hasil 100%.

3.2.5.3 Pemeriksaan Correctness

Setiap item kebutuhan diperiksa dengan melakukan query terhadap subjek dengan predikat dan objek sebagai acuan yang kemudian menghasilkan kalimat kebutuhan yang tersedia pada triple ontology. Jika hasil query tidak menunjukkan hasil apapun, maka kebutuhan tersebut tidak benar. Daftar query correctness ditunjukkan pada Gambar 3.10.

✓ QueryCorrectnessSPPK-3-001	login_as(?S, pegawai) -> sqwrlselect(?S, login_as, pegawai)
✓ QueryCorrectnessSPPK-3-002	login_as(?S, penilai) -> sqwrlselect(?S, login_as, penilai)
✓ QueryCorrectnessSPPK-3-003	melihat(?S, Biodata_pegawai) -> sqwrlselect(?S, melihat, Biodata_pegawai)
✓ QueryCorrectnessSPPK-3-004	mengubah(?S, Biodata_pegawai) -> sqwrlselect(?S, mengubah, Biodata_pegawai)
✓ QueryCorrectnessSPPK-3-005	melihat(?S, Daftar_penilaian) -> sqwrlselect(?S, melihat, Daftar_penilaian)
✓ QueryCorrectnessSPPK-3-006	menambah(?S, SKP) -> sqwrlselect(?S, menambah, SKP)
✓ QueryCorrectnessSPPK-3-007	melihat(?S, Hasil_penilaian_skp) -> sqwrlselect(?S, melihat, Hasil_penilaian_skp)
✓ QueryCorrectnessSPPK-3-008	mencetak(?S, SKP) -> sqwrlselect(?S, mencetak, SKP)
✓ QueryCorrectnessSPPK-3-009	melihat(?S, Hasil_ppk) -> sqwrlselect(?S, melihat, Hasil_ppk)
✓ QueryCorrectnessSPPK-3-010	mencetak(?S, PPK) -> sqwrlselect(?S, mencetak, PPK)
✓ QueryCorrectnessSPPK-4-012	melihat(?S, Biodata_penilai) -> sqwrlselect(?S, melihat, Biodata_penilai)
✓ QueryCorrectnessSPPK-4-013	mengubah(?S, Biodata_penilai) -> sqwrlselect(?S, mengubah, Biodata_penilai)
✓ QueryCorrectnessSPPK-4-014	melihat(?S, List_pegawai) -> sqwrlselect(?S, melihat, List_pegawai)
✓ QueryCorrectnessSPPK-4-015	melihat(?S, Hasil_penilaian_skp_pegawai) -> sqwrlselect(?S, melihat, Hasil_penilaian_skp_pegawai)
✓ QueryCorrectnessSPPK-4-016	melihat(?S, Hasil_ppk_pegawai) -> sqwrlselect(?S, melihat, Hasil_ppk_pegawai)
✓ QueryCorrectnessSPPK-4-017	menghitung(?S, Penilaian_skp) -> sqwrlselect(?S, menghitung, Penilaian_skp)
✓ QueryCorrectnessSPPK-5-018	menghitung(?S, PPK) -> sqwrlselect(?S, menghitung, PPK)
✓ QueryCorrectnessSPPK-5-020	menambah(?S, Data_pegawai) -> sqwrlselect(?S, menambah, Data_pegawai)
✓ QueryCorrectnessSPPK-5-021	menambah(?S, Data_penilai) -> sqwrlselect(?S, menambah, Data_penilai)

Gambar 3.10 Daftar Query Correctness

Gambar 3.11 merupakan contoh hasil dari query correctness yang didasarkan oleh SPPK-3-006.

SQWRL Queries	OWL 2 RL	QueryCorrectnessSPPK-3-005	QueryCorrectnessSPPK-3-006
S	C1	C2	
Pegawai	menambah	SKP	

Gambar 3.11 Hasil Query Correctness oleh SPPK-3-006

Correctness : $|F_{\text{item}}| / |R_{\text{item}}| = 19/19 = 100\%$

Dari hasil query yang diperoleh menunjukkan jumlah item kebutuhan yang telah terpetakan (telah diperiksa kebenarannya) menjadi triple di dalam sistem ontologi adalah 19 dan jumlah semua item kebutuhan dalam spesifikasi kebutuhan adalah 19 yang kemudian dimasukkan kedalam rumus correctness sehingga mendapatkan hasil 100%.

4. KESIMPULAN

Dalam penelitian rekayasa kebutuhan dengan metode linguistik dan ontologi pada sistem penilaian prestasi kerja pegawai dinas kominfo malang, beberapa hal yang dapat disimpulkan yaitu :

1. Pemodelan spesifikasi kebutuhan menggunakan DITA menghasilkan file DITA beserta strukturnya, dengan setiap file terdapat referensi terhadap Pemodelan Ontologi. Pemodelan Ontologi berisi daftar spesifikasi kebutuhan dalam bentuk triple ontology yang setiap subjeknya memiliki relasi predikat terhadap masing-masing objek.

2. Validasi dan verifikasi kebutuhan dilakukan dengan cara pemeriksaan *completeness*, *consistency* dan *correctness*. Pemeriksaan tersebut menggunakan proses *query* terhadap *triple ontology* yang kemudian disesuaikan secara manual terhadap *triple* kebutuhan. Hasil dari pemeriksaan kebutuhan tersebut menunjukkan bahwa nilai dari masing-masing *completeness*, *consistency* dan *correctness* adalah 100%.
- Siddiqui, F., & Alam, M., A. 2011. *An Ontology Based Approach for Requirement Incoherence Detection*. Department of Computer Science, Hamdard University, New Delhi, India.
- Sommerville, I. 2005. *Software Engineering*(7th Edition). Palgrave Macmillan.
- Swiss, Q. C. 2012. *SwissQ Requirements Trends & Benchmarks Switzerland*

DAFTAR PUSTAKA

- Akmajian, A., Demers, R., Farmer, A., Harnish, R. 2010. *Linguistics: An Introduction to Language and Communication*. Cambridge, MA: The MIT Press. ISBN 0-262-51370-6.
- Alspaugh, T. A., & Antón, A. I. 2008. *Scenario Support for Effective Requirements*. *Information and Software Technology*, 50(3), 198-220.
- Day, D., Priestley, M., & Schell, D. 2005. *Introduction to the Darwin Information Typing Architecture*. IBM corporation.
- Dessler, Gary, 1997, *Manajemen Sumber Daya Manusia*. Edisi ke-7, Alih bahasa, Jilid 1 & Jilid 2, Prenhallindo, Jakarta.
- Hoffman, H., & Lehner, F. 2001. *Requirements Engineering as a Success Factor in Software Projects*. IEEE Software.
- Hu, H., Zhang, L., Ye, C. 2010. *Semantic-based Requirements Analysis and Verification*. *International Conference on Electronics and Information Engineering (ICEIE 2010)*.
- Laplante, Phillip. 2007. *What Every Engineer Should Know about Software Engineering*. Boca Raton: CRC. ISBN 978-0-8493-7228-5. Retrieved 2011-01-21.
- Lautenbacher, F., Bawer, B., Sieber, T., Cabral, A. 2010. *Linguistics-Based Modeling Methods and Ontologies in Requirements Engineering*. *International Journal of Enterprise Information Systems*.
- Rouibah, K., & Al-Rafee, S. 2009. *Requirement Engineering Elicitation Methods: A Kuwaiti Empirical Study About Familiarity, Usage and Perceived Value*. *Information Management & Computer Security*, 17(3), 192-217.