

MODEL KNOWLEDGE MANAGEMENT SYSTEM PADA LEMBAGA RISET BERBASIS SEMANTIC WEB DAN WEB 2.0

Pita Larasati Fauziah Nur¹, Dana Indra Sensuse¹, dan Dwi Handoko²

¹Fakultas Ilmu Komputer, Universitas Indonesia, Kampus Baru UI Depok, Jawa Barat, 16424, Indonesia

²Bidang Sistem Informasi dan Komputasi, Pusat Teknologi Informasi dan Komunikasi, Badan Pengkajian dan Penerapan Teknologi, Jakarta, 10340, Indonesia

E-mail: pila05@ui.edu

Abstrak

Bermula dari kebutuhan lembaga riset untuk mengelola pengetahuan dengan sumber utama berupa *human capital*, BPPT memerlukan suatu sistem yang dapat mengakomodasinya. Pengembangan sistem yang lebih dikenal dengan *knowledge management system* bertujuan untuk *sharing* data, informasi, dan pengetahuan serta komunikasi yang terjadi pada lembaga tersebut. Model berupa prototipe sistem yang berbasis *web* ini menggunakan teknologi *semantic web* dengan penerapan ontologi serta menggunakan *web 2.0* sebagai media untuk kolaborasi antara sesama *knowledge worker*. Model *knowledge management system* yang dihasilkan dari pengembangan ini adalah *knowledge sharing system* yang terdiri dari portal *semantic web* serta *blog* yang masih bersifat *independent*.

Kata Kunci: *knowledge management system, semantic web, ontologi, BPPT*

Abstract

Starting from research institutions need to manage knowledge with the main source of human capital, BPPT requires a system that can accommodate it. Development system that is better known as knowledge management system aimed at sharing data, information, and knowledge as well as the communication that occurs in these institutions. The model that is a prototype of webbased system uses semantic web technology with the application of ontologies and uses web 2.0 as a medium for collaboration among knowledge workers. Model of knowledge management system resulting from this development is the knowledge sharing system that consists of semantic web portals and blogs that are still independent.

Keywords: *knowledge management system, semantic web, ontology, BPPT*

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi membuat beberapa sektor di Indonesia khususnya lembaga riset memerlukan sarana yang memadai agar proses tukar menukar data, informasi, dan pengetahuan dapat berjalan dengan baik. Seperti yang telah diketahui, suatu lembaga riset menggunakan *human capital* sebagai sumber pengetahuan, inovasi, dan pembaharuan. Lembaga riset biasanya bergantung kepada komunikasi informal antara individu dan bersifat *people-centric operations* [1]. Hal ini yang menyebabkan seringkali penyebaran pengetahuan terputus sehingga informasi-informasi penting tidak dapat tersampaikan dengan baik antara sesama individu pada lembaga tersebut. Oleh karena itu, perlu dikembangkan sebuah sistem yang berperan dalam membantu pengelolaan pengetahuan atau

lebih dikenal dengan sebutan *knowledge management system*.

Motivasi. Beberapa lembaga riset dunia, salah satunya NASA, telah lama mengembangkan *knowledge management system* dengan basis *web*. Penggunaan *web* sebagai media dikarenakan sarana yang mudah, serta cepat dalam mengakses data, informasi, dan pengetahuan. Lembaga ini menggunakan teknologi *semantic web* untuk mengembangkan *knowledge management system* miliknya.

Ontologi merupakan ide dasar dari pemrosesan data (*machine-readable system*) pada *semantic web*. Ontologi merupakan suatu gambaran tentang konsep dan relasi antar konsep yang dapat bertujuan untuk *knowledge sharing* dan *knowledge reuse* [2]. Dengan melihat perkembangan tersebut, Badan Pengkajian dan Penerapan Teknologi (BPPT) yang merupakan salah satu lembaga riset di Indonesia pun

memerlukan *knowledge management system* sebagai sarana atau media komunikasi, kolaborasi, penyebaran data, informasi, serta pengetahuan agar hambatan-hambatan yang umum terjadi pada lembaga riset ini dapat teratasi.

Tujuan umum dikembangkannya *knowledge management system* di lembaga riset BPPT ini adalah untuk mengetahui model *knowledge management system* seperti apa yang cocok dikembangkan dan dijalankan. Selain itu, tujuan khusus pengembangan yang merupakan *requirement* pada sistem, di antaranya adalah mendukung pekerjaan atau proyek tim yang terdistribusi, mendukung komunikasi antara individu, *sharing* informasi dan pengetahuan berdasarkan proyek yang dikembangkan di BPPT.

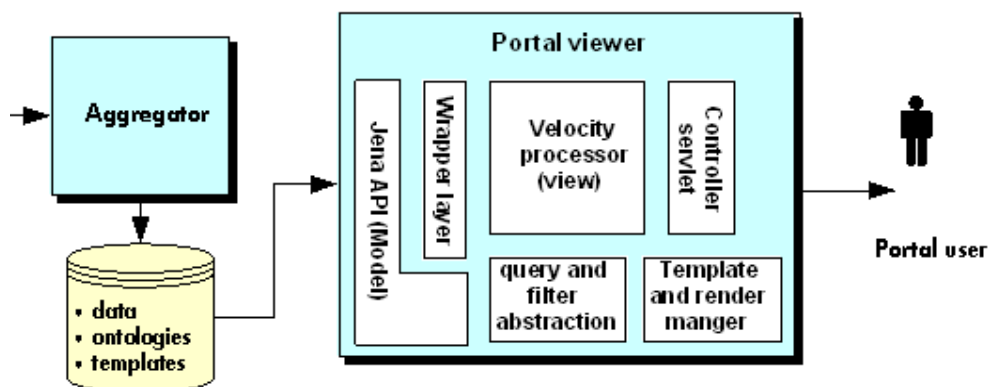
Ontologi diambil dari istilah filsafat, yaitu suatu penjelasan sistematis dari suatu yang *exist*. Lalu, pengertian tersebut diganti menjadi suatu objek, konsep, dan entitas lain yang diasumsikan "*exist*" pada suatu area dan terdapat relasi di antaranya. Ontologi menjelaskan secara eksplisit suatu konsep. Konsep ini mengacu kepada kumpulan konsep yang digunakan untuk merepresentasikan suatu objek nyata atau suatu *knowledge*. Eksplisit artinya mendefinisikan tipe konsep dan batasan yang digunakan dalam memberikan arti formal yang dipahami oleh bahasa mesin. Dalam konteks *artificial intelligence* (AI), dapat dideskripsikan ontologi adalah suatu mekanisme pendefinisian sebuah terminologi representasi. Pada ontologi, bentuk representasi primitif berupa *classes* atau *set*, *attributes* atau *properties*, dan *relationship* (atau relasi antara *class* dan *member*) [2].

Komunitas peneliti AI bersama-sama mencari suatu cara untuk membantu *sharing knowledge*. Mereka menginginkan suatu program komputer dapat berinteraksi dengan dan berdiri pada informasi dari program komputer lainnya. Akhirnya, pada tahun 2001 Tim Berners-Lee membuat misi menjadi nyata dengan menciptakan

semantic web. Ide dasar dari *semantic web* adalah *web* yang memiliki data yang terdefinisi dan saling terhubung sehingga data tersebut dapat digunakan untuk penemuan, otomasi, integrasi, dan *reuse* secara efektif melalui beberapa aplikasi. Ontologi ini yang merupakan ide utama pemrosesan data oleh mesin (atau disebut juga *machine-readable*) pada *semantic web* [3].

Semantic web diekspresikan dalam spesifikasi formal, yaitu *Resource Description Framework* (RDF). Beberapa format *data interchange* (misal RDF/XML, N3, Turtle, N-Triples), dan notasi seperti RDF Schema (RDFS) dan *the Web Ontology Language* (WOL), semua menampilkan deskripsi formal suatu konsep, terminologi, dan relasi di antara domain pengetahuan yang diberikan.

Previous Works, beberapa penelitian atau proyek menggunakan teknologi *semantic web* dalam pengembangan *knowledge management system* yang telah dilakukan sebelumnya. Penelitian tersebut diantaranya, yaitu *Semantic Organizer* milik NASA [4], *Organik Project* yang dikembangkan oleh Uni Eropa dengan integrasi teknologi *semantic web* dan *enterprise software 2.0*, *VTT's ICT Knowledge Portal* [5] yang berasal dari Finlandia yang memiliki ide berupa suatu ontologi dapat mengelola informasi tentang *project*, *people*, *document*, dan *product*, serta SWED (*Semantic web Environmental Directory*) yang merupakan suatu proyek yang dibiayai oleh Uni Eropa sebagai bagian dari proyek SWAD-E (*Semantic web Advanced Development for Europe*) yang bertujuan untuk mendukung W3C *Semantic web* di Eropa, menyediakan penelitian, demonstrasi, dan memastikan teknologi *semantic web* mencapai target yang lebih baik ke dalam *networked computing*. SWED ini berupa *open source* yang digunakan untuk mengembangkan *knowledge management system* di BPPT. SWED memiliki struktur portal seperti terlihat pada gambar 1 [6].



Gambar 1. Struktur portal SWED.

Komponen utama adalah tampilan portal. Tampilan portal ini berjalan sebagai aplikasi *web* pada Java Servlet *container* dan menyediakan *web interface* pada *semantic web* data yang ada di portal tersebut. Lalu, komponen yang kedua adalah *aggregator* yang secara periodik memindai daftar dari sumber situs yang diketahui dan mengunggah tiap perubahan RDF data ke portal *database* sehingga dapat ditampilkan oleh *viewer*.

Meskipun dalam diagram di atas memperlihatkan bahwa terdapat satu *database*, akan tetapi kenyataannya adalah portal mengambil data dari *multiple files* yang dimasukkan ke dalam memori, seperti layaknya dari sebuah *database* tambahan. Tampilan *templates* berupa *file* sederhana (*local* atau *retrieval* via HTTP URLs) dan dimasukkan serta dikelola oleh *template engine*. Ontologi merupakan *local RDF* (RDFS atau OWL) *files* dan dimasukkan ke dalam memori. Biasanya data disimpan di dalam sebuah *database*, tetapi dalam portal sederhana ini disimpan ke dalam memori dari *static file*.

Tampilan portal mengadopsi desain MVC (*Model-View-Controller*). Model ditampilkan oleh sebuah *linked* kumpulan dari Java *classes*. *Instance* portal dispesifikasikan oleh *Datasource* yang memberikan akses untuk representasi *filter state*, kumpulan *facets*, *datasource*, dan RDF API yang disediakan oleh Jena Library. Lalu, komponen *View* menggunakan Jakarta *Velocity template engine* untuk membuat *displayed resource* berdasarkan kepada kumpulan tampilan *templates*. *Controller* adalah Java Servlet untuk membangun dan memanggil *velocity template*. Aplikasi *web* portal berjalan pada JSP *container* seperti Tomcat [6].

2. Metodologi

Metodologi penelitian ini merupakan studi kasus pada lembaga riset BPPT (Badan Pengkajian dan Penerapan Teknologi). Adapun lima tahapan yang dilalui BPPT. Pertama, melakukan studi kasus terhadap beberapa lembaga riset yang telah mengembangkan *knowledge management system* untuk lembaga itu sendiri ataupun proyek penelitian. Tahap kedua, analisis *project flow* dan struktur organisasi di BPPT serta penilaian berdasarkan *contingency factor*. Ketiga, analisis dan desain ontologi untuk domain riset sesuai dengan pemahaman terhadap *project flow* di BPPT. Keempat, mencari *open source* yang dapat mengakomodasi kebutuhan sistem untuk dikembangkan di BPPT dengan melihat teknologi dan fitur yang dimilikinya. Kemudian, desain prototipe *knowledge management system*

berdasarkan *project flow* BPPT dengan menggunakan *open source* yang telah dipilih. Tahapan terakhir yaitu uji fungsionalitas terhadap sistem yang telah dikembangkan.

3. Analisis dan Pembahasan

Sebelum membahas tahapan pengembangan portal, terdapat pertimbangan dalam penggunaan *open source* untuk *knowledge management system* di BPPT. Pertimbangan-pertimbangan berikut pun berperan dalam pemenuhan *requirement* yang telah dijelaskan pada tujuan pengembangan sistem, yaitu *Component Reuse* dan *Blog Wordpress*.

Pada saat implementasi, sistem ini tidak dikembangkan dari awal (*built from scratch*), tapi dengan menggunakan *component reuse*. Komponen tersebut merupakan *open source* yang bernama *portalCore* yang telah digunakan pada proyek Uni Eropa, yaitu SWED (telah dibahas pada *previous works*). *Component reuse* *portalCore* ini melalui beberapa pertimbangan.

Sebelum memilih *portalCore*, terdapat satu *open source* lain yaitu *Cyn.in*. Akan tetapi, sistem ini hanya merupakan *enterprise 2.0* yang merujuk kepada *web 2.0* (*social software*) tetapi tidak menggabungkan *semantic web* dalam implementasi. Sedangkan untuk mengembangkan sistem di BPPT yang menitikberatkan pada *sharing knowledge*, *knowledge management system* yang diimplementasikan berjenis *knowledge sharing system* di mana penerapannya dengan menggunakan teknologi *semantic web*. Implementasi sistem di BPPT pun tidak sampai kepada produk akhir, hanya sebatas prototipe yang berdasarkan kepada penggunaan *component reuse* dan prototipe desain GUI. Akhirnya pencarian *open source* dilanjutkan dengan memilih *portalCore*.

Seperti yang telah diketahui *portalCore* ini menggunakan teknologi *semantic web*, dan pada situsnya pun dijelaskan bahwa sangat cocok dalam mendemokan prototipe sistem. *Component reuse* ini berfungsi dalam implementasi *core* atau inti dari sistem yang dikembangkan di BPPT. *Core* ini berupa sistem informasi yang ditampilkan pada sistem BPPT Intranet.

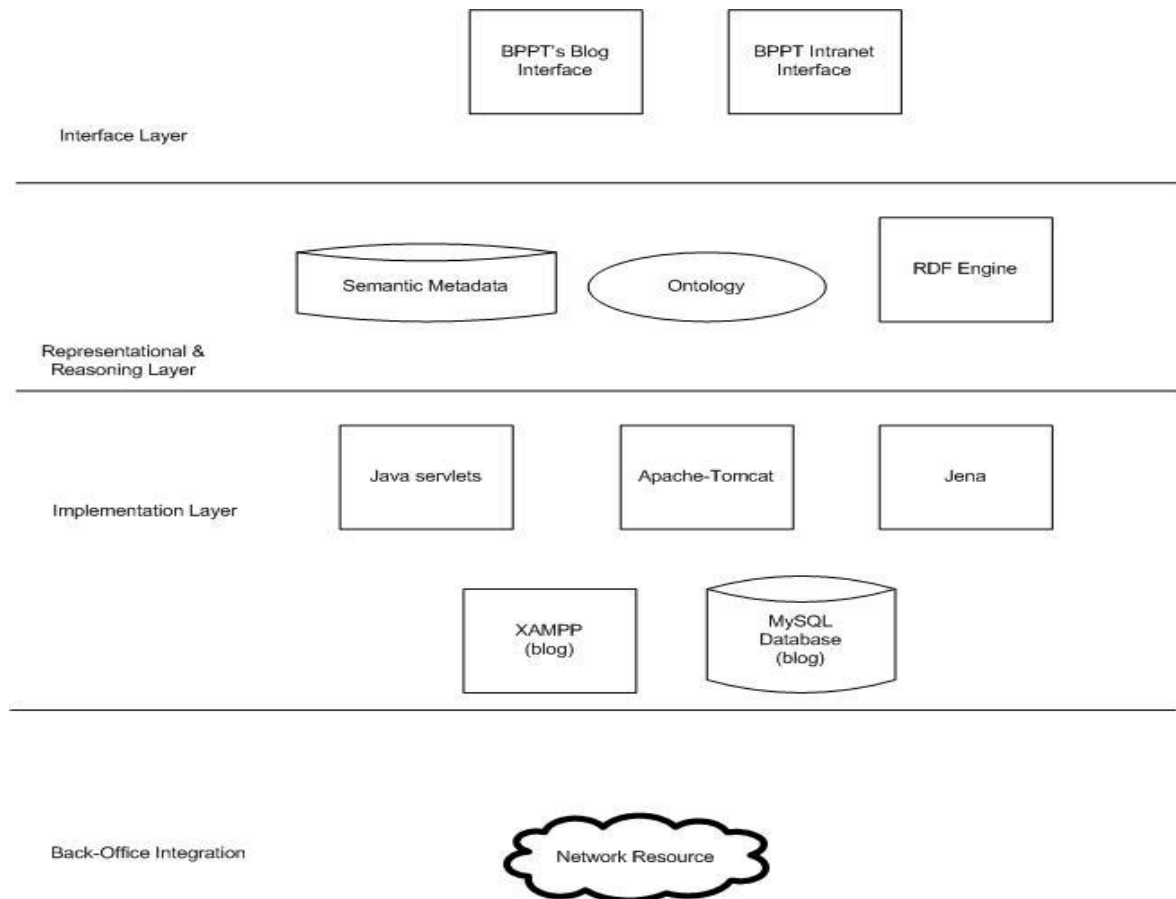
Blog Wordpress berfungsi sebagai prototipe dalam memenuhi *requirement* tujuan pengembangan sistem, yaitu mendukung komunikasi antar sesama *knowledge worker* di BPPT. *Blog* pada sistem BPPT Intranet masih bersifat *independent*, artinya *blog* tidak diintegrasikan dalam sistem BPPT Intranet tetapi dihubungkan dengan *link*. Pengembangan model

knowledge management system di BPPT melalui beberapa tahapan analisis dan desain serta implementasi. Adapun tahapan analisis meliputi analisis *project flow* BPPT dengan pengamatan dan wawancara kepada salah satu *knowledge worker* yang kemudian diikuti dengan penilaian tambahan dengan melakukan analisis terhadap kultur organisasi BPPT (*contingency factor*).

Arsitektur Sistem BPPT Intranet, *interface layer* merupakan tampilan *knowledge management system* milik BPPT, yang terdiri dari *blog* dan BPPT Intranet *interface*. Terlihat pada gambar 2, BPPT Intranet memiliki *representational & reasoning layer* yang berisikan ontologi untuk menyimpan *class* dan *link types* serta *rules*, *semantic metadata* sebagai *repositories* untuk menyimpan *link*, *atributes value*, dan *instance* atau *item*. *RDF engine* berfungsi untuk menjalankan *rules* yang telah dibuat. Pada *implementation layer*, Java Servlet

digunakan untuk navigasi antara *interlink item* yang dibuat dengan *repository* untuk melihat *files*, memasukkan *metadata*, *search item* tertentu. Sistem BPPT Intranet termasuk komponen *inference* dibuat di atas *framework* Jena dan sistem ini dapat berjalan pada Apache-Tomcat. Sedangkan XAMPP digunakan untuk menjalankan *blog* BPPT dengan MySQL sebagai tempat menyimpan *database blog*. Terakhir, *back-office integration* yang terdiri dari *network resource* agar sistem tersebut dapat diakses.

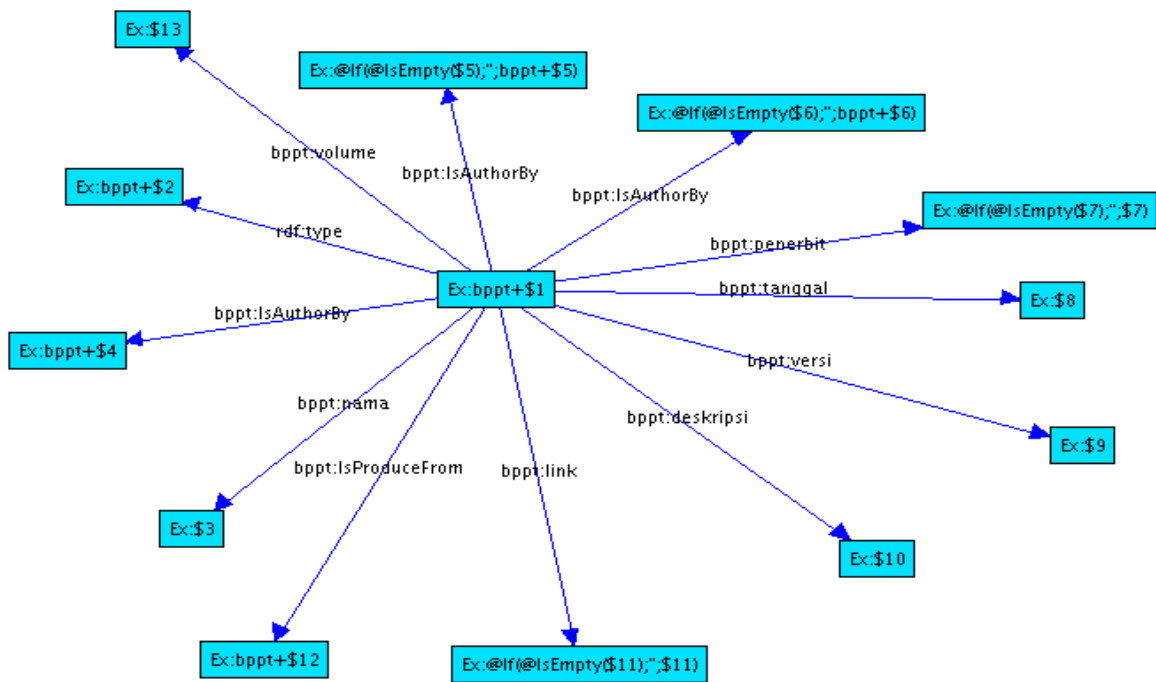
Desain ontologi pada BPPT Intranet tidak menggunakan *ontology reuse*, tetapi membuat sendiri dengan memperhatikan *project flow* yang terdapat di lembaga riset BPPT. Desain dibuat dengan menggunakan *tool* Protégé 3.4 yang terdiri dari lima *classes* utama, yaitu *Project*, *Product*, *Competency*, *Person*, dan *Organization* dengan menggunakan bantuan OntoVizTab, bisa dilihat pada gambar 3.



Gambar 2. Arsitektur BPPT Intranet.



Gambar 3. Classes Ontologi BPPT.



Gambar 4. Map graph class Product.

Persiapan Instance Data, setelah membuat desain ontologi BPPT, hal yang dilakukan adalah menyiapkan instance data. Dalam portalCore, instance data ditampilkan menggunakan format RDF dengan sintaks .N3. Pertama-tama, instance disimpan dalam file dengan format spreadsheet (.xls) yang diubah kembali formatnya menjadi (.csv), kemudian dengan menggunakan RDF123 yang berfungsi sebagai converter ke dalam format RDF. Akan tetapi, terlebih dahulu dibuat mapping graph menggunakan RDF123 sehingga hasilnya disimpan dalam output berupa .N3.

Berdasarkan ontologi BPPT yang telah dibuat, terdapat lima class utama yang seluruhnya akan dibuatkan instance data, yaitu class Person, Project, Product, Competency, dan Organization sehingga akan ada lima file dengan tipe .csv. Kelima file ini berisikan atribut-atribut yang ditampilkan dalam suatu tabel. Masing-masing atribut inilah yang akan diisi dengan instance data. Tiap satu baris pada tabel menghasilkan satu instance data. Kemudian, dibuatlah map graph dengan menggunakan RDF123. Gambar 4 merupakan salah satu contoh map graph untuk class Product.

Pengerjaan map graph tentunya memiliki format tertentu dalam penggunaan namespace, yaitu Ex:, \$n, logical expression @If(B;E;E) dan @IsEmpty(E). Map graph disimpan dengan tipe file .xgmml dan dapat di-convert menjadi format RDF. Konversi dilakukan dengan memilih menu

”Output” pada RDF123 dan diikuti dengan memilih ”View SpreadSheet in RDF”. Lalu, simpan file dengan memilih ”Display type: N3” [2].

Pendefinisian rules bertujuan untuk proses inference disimpan dalam file bernama ”BPPT.rules”. Selain ontologi dan instance data, rule juga merupakan input pada portal BPPT Intranet. Pada portalCore, terdapat RDFS Closure Rules yang telah didefinisikan. Beberapa rules yang telah terdefiniskan dapat dilihat pada gambar 5. rdfs2 menjelaskan bahwa suatu relasi p menghubungkan x dengan y, jika relasi p memiliki batasan domain c, maka x instance dari c. rdfs3 menjelaskan bahwa suatu relasi p menghubungkan x dengan y, jika relasi p memiliki batasan range c, maka x instance dari c. rdfs5a menjelaskan bahwa jika a subproperty dari b, dan b subproperty dari c, maka a juga subproperty dari c.

rdfs5b menjelaskan bahwa suatu property adalah subproperty dari dirinya sendiri. rdfs6 menjelaskan bahwa suatu relasi p menghubungkan a dengan b, jika relasi p subproperty dari q, maka relasi q pun menghubungkan a dengan b. rdfs7 menjelaskan bahwa suatu class adalah subclass dari dirinya sendiri. rdfs8 menjelaskan bahwa jika a subclass b, dan b subclass c, maka a juga subclass c. rdfs9 menjelaskan bahwa jika x subclass y, a instance x, maka a juga instances y.

```
[rdfs2: (?x ?p ?y), (?p rdfs:domain
?c) -> (?x rdf:type ?c)]
[rdfs3: (?x ?p ?y), (?p rdfs:range
?c) -> (?y rdf:type ?c)]
[rdfs5a: (?a rdfs:subpropertyOf
?b), (?b rdfs:subpropertyOf ?c) ->
(?a rdfs:subpropertyOf ?c)]
[rdfs5b: (?a rdf:type rdf:Property)
-> (?a rdfs:subpropertyOf ?a)]
[rdfs6: (?a ?p ?b), (?p
rdfs:subpropertyOf ?q) -> (?a ?q
?b)]
[rdfs7: (?a rdf:type rdfs:Class) ->
(?a rdfs:subclassOf ?a)]
[rdfs8: (?a rdfs:subclassOf ?b),
(?b rdfs:subclassOf ?c) -> (?a
rdfs:subclassOf ?c)]
[rdfs9: (?x rdfs:subclassOf ?y),
(?a rdf:type ?x) -> (?a rdf:type
?y)]
```

Gambar 5. RDFS Closure Rules.

Pada pengembangan untuk BPPT sendiri, terdapat *rules* yang didefinisikan khusus. Gambar 6 merupakan salah satu contoh *rule*. *Rule* ini menjelaskan bahwa jika A mengerjakan proyek B, dan proyek B dikembangkan oleh bidang C, maka bidang C tersebut dijalankan oleh A.

```
(?A bppt:IsDoing ?B), (?B
bppt:DevelopedBy ?C) -> (?C
bppt:IsWorkOnBy ?A).
(?X bppt:IsWorkOnBy ?Y) -> (?Y
bppt:WorksOn ?X) .
```

Gambar 6. Rule untuk menyatakan seseorang bekerja pada bidang tertentu.

Konfigurasi yang dilakukan dalam penggunaan *component reuse* portalCore untuk pengembangan sistem BPPT terdiri dari tiga tahapan, yaitu pendefinisian *data source*, *facets*, dan *templates*. Pendefinisian *data source* dapat dilihat dalam WEB-INF/config/sources.n3 yaitu sebuah *file* RDF dengan sintaks N3 [7]. Pada konfigurasi *data source* (gambar 7) membutuhkan beberapa data. Data dikategorikan ke dalam tiga kelompok, yaitu ontologi, *instance* data, dan *closure rule*.

Sedangkan pendefinisian *facet* digunakan untuk navigasi data. Spesifikasi *facet* menggunakan *pcv:facet property* sedangkan *pcv:linkProp* untuk menampilkan objek yang dicari berdasarkan *property*. Terdapat dua *subclass* dari *pcv:Facet*, yaitu

pcv:HierarchicalFacet (*resources* yang terindeks diklasifikasikan dengan pengaturan sktruktur hirarki dan *pcv:AlphaRangeFacet* (*resources* dikelompokkan berdasarkan huruf pertama dari *literal-valued property*). Sebuah *facet* yang dideklarasikan hanya *pcv:Facet* akan berfungsi sebagai *flat facet* yang akan menampilkan *instance* yang dihubungkan oleh sebuah *property* [7]. Gambar 8 merupakan salah satu contoh sintaks dalam pendefinisian *facet Project*.

```
pcv:sourceURL
<portal://data/personbppt.n3> ;
pcv:sourceURL
<portal://data/orgbppt.n3> ;
pcv:sourceURL
<portal://data/productbppt.n3> ;
pcv:sourceURL
<portal://data/competencybppt.n3> ;
pcv:sourceURL
<portal://data/projectbppt.n3> ;
pcv:ontologySourceURL
<portal://data/ontologyBPPT.owl> ;
pcv:closureRulesURL
<portal://data/BPPT.rules> ;
```

Gambar 7. Pendefinisian *data source*.

```
.....
bppt:p2titleFacet a
pcv:AlphaRangeFacet;
rdfs:label "Title" ;
pcv:linkProp bppt:nama;
pcv:order "1"^^xsd:integer;
.

bppt:p2typeFacet a
pcv:HierarchicalFacet;
rdfs:label "Type";
pcv:linkProp rdf:type;
pcv:widenP rdfs:subclassOf ;
pcv:facetBase bppt:Project ;
pcv:order "2"^^xsd:integer;
.
```

Gambar 8. Contoh sintaks dalam pendefinisian *facet project*.

Tahapan terakhir konfigurasi portal yaitu pendefinisian *templates*. *Templates* ini digunakan untuk memungkinkan tiap *data source* menggunakan *template* yang berbeda untuk setiap fungsi. Pendefinisian *templates* sebanyak jumlah *class* utama, yaitu *Person*, *Project*, *Product*, *Competency*, dan *Organization*. Gambar 9 adalah contoh tampilan pendefinisian *templates*.

```

pcv:template [a pcv:Template;
pcv:templateContext "page" ;
pcv:templatePath
<portal://templates/pagePersonBPPT
.vm> ;
pcv:templateClass
bppt:Person;
];

.....
pcv:template [a pcv:Template;
pcv:templateContext "page" ;
pcv:templatePath
<portal://templates/pageCompetency
BPPT.vm> ;
pcv:templateClass
bppt:Competency;
];

```

Gambar 9. Tampilan pendefinisian *templates*.

Visualisasi *Templates*, pada tahap visualisasi *templates*, *portalCore* membagi tampilan menjadi tiga, yaitu *browse page*, *results page*, dan *resource page*. *Browse page* merupakan halaman yang menampilkan *facets* dari suatu *class*, *results page* adalah halaman hasil pemilihan *browse-facets*, sedangkan *resources page* menampilkan deskripsi dari suatu *resource*. Modifikasi *code* dilakukan pada *resource page* [8]. Salah satu bagian dari *code* pada *resource page* yang telah dimodifikasi dapat dilihat pada gambar 10.

```

<tr> <td width="10%"><b> Status:
</b> </td>
<td width="90%">
    $resource.getPropertyValue
    ("bppt:statusproyek")
    </td>
</tr>
<tr height="8"></tr>

```

Gambar 10. Salah satu bagian dari *code* pada *resource page* yang telah dimodifikasi.

Pada *code* di atas, *statement* yang akan ditampilkan berupa tabel. Terdapat *statement* yang menyatakan adanya tambahan satu baris pada tampilan *resources page*, yaitu *Status*. Pada bagian yang berisi atribut *Status*, jika *resource* memiliki *property* "status proyek", maka status akan ditambahkan dalam satu baris.

Hasil prototipe berdasarkan *existing framework* yang telah diimplementasikan dan prototipe yang masih berdasarkan desain GUI dapat dijelaskan seperti pada tabel I yang berisikan pemetaan antara tujuan pengembangan *knowledge management system* dengan *use cases* yang didesain serta ketersediaan desain GUI dan implementasi. Dapat disimpulkan masih terdapat *use case* yang belum diimplementasikan, yaitu *use case update* (menambah, mengubah, dan menghapus) data *person*, *project*, *product*, dan *competency*. *Use case* tersebut masih berupa prototipe desain GUI. Gambar 11 dan gambar 12 merupakan contoh tampilan halaman utama BPPT Intranet beserta *blog*.

Setelah tahapan implementasi, dilakukan uji fungsionalitas yang bertujuan untuk mengetahui apakah semua fitur-fitur yang ada telah berfungsi dengan baik atau belum. Dari hasil uji fungsionalitas, didapat bahwa fitur *update* otomatis belum diimplementasikan sehingga dapat dikatakan belum berfungsi.

4. Kesimpulan

Pada penilaian berdasarkan *contingency factor*, model *knowledge management system* yang cocok dikembangkan di BPPT, adalah *knowledge capture* dan *sharing system*. Ini dapat dilihat dari nilai kumulatif yang terbanyak pada *internalization* dan *exchange*. Akan tetapi, kami mengambil kesimpulan bahwa *knowledge sharing* merupakan yang tepat dikarenakan pengamatan berdasarkan tujuan dari pengembangan *knowledge management system* itu sendiri. Selain itu, model *knowledge sharing system* yang dipilih pun berdasarkan atas *previous works*, analisis *project flow*, serta *contingency factor*.

Saran untuk penelitian selanjutnya adalah optimalisasikan terhadap implementasi *use case* yang sudah ada, integrasi *blog* dengan BPPT Intranet sehingga pengguna yang ingin mencari arsip proyek A, dapat langsung pula mendapatkan artikel yang terkait dengan proyek A tersebut. Pengguna yang ingin *posting* artikel terkait proyek, artikel tersebut langsung membuat *link* terhadap data proyek yang sesuai. Selain itu pengembangan *collaborative software* lainnya, yaitu Wiki. Hal yang terakhir untuk pengembangan selanjutnya adalah BPPT Intranet tidak hanya membantu internal *knowledge worker* di lembaga riset ini, tetapi juga pengetahuan yang disimpan di dalam sistem ini dapat dimanfaatkan oleh masyarakat.



Gambar 11. Halaman utama.



Gambar 12. Halaman *blog*.

TABEL I
HASIL PROTOTYPE BPPT INTRANET

Tujuan KMS	Use Cases	Desain GUI	Implementasi
Mendukung pekerjaan atau proyek tim yang terdistribusiterdistribusi	• <i>Use case</i> mencari arsip <i>project</i>	Ya	Ya
	• <i>Use case</i> mencari dokumen berdasarkan <i>person</i>	Ya	Ya
	• <i>Use case</i> mencari <i>person</i> berdasarkan <i>skill</i>	Ya	Ya
Mendukung komunikasi antara individu <i>Sharing</i> informasi dan pengetahuan berdasarkan proyek yang dikembangkan di BPPT.	• <i>Use case</i> menyebarkan artikel bebas	Ya	Ya
	• <i>Use case</i> menyebarkan artikel terkait proyek	Ya	Ya
	• <i>Use case</i> mencari dokumen terkait <i>current project</i> berdasarkan <i>project name</i>	Ya	Ya
	• <i>Use case update</i> (menambah, mengubah, dan menghapus) data <i>person, project, product, dan competency</i> .	Ya	Tidak

Referensi

- [1] Organik Knowledge Management, Objective, Organik Project, <http://organik-project.eu/eng/objectives.asp>, 2008, retrieved March 16, 2009.
- [2] T. Gruber, *Ontology*, Springer-Verlag, <http://tomgruber.org/writing/ontology-definition-2007.htm>, 2009, retrieved May 31, 2009.
- [3] T. Gruber, "Ontology of Folksonomy: a Mash-up of Apples and Oranges," *International Journal on Semantic Web and Information Systems*, vol.2, pp. 1-11, 2007.
- [4] R.M. Keller, D.C. Berrios, R.E. Carvalho, D.R. Hall, J. Rich, I.B. Sturken, K.J. Swanson, & S.R. Wolfe, "Semantic Organizer: a Customizable Semantic Repository for Distributed NASA Project Teams" *In International Semantic Web Conference ISWC*, pp. 767-781, 2004.
- [5] A. Bäck, S. Vainikainen, C. Södergård, & H. Juhola, *Semantic Web Technologies in Knowledge Management, Scientific Information Exchange*, <http://elpub.scix.net/data/works/att/0334.content.pdf>, retrieved March 10, 2009.
- [6] SWAD Europe, SWAD-E Portal Structure, Hewlett-Packard Development Company, <http://www.swed.org.uk/swed/doc/portal-structure.html>, 2004, retrieved June 3, 2009.
- [7] SWAD Europe, SWAD-E Portal Customization, Hewlett-Packard Development Company, <http://www.swed.org.uk/swed/doc/portal-customization.html>, 2004, retrieved June 3, 2009.
- [8] Y. Aprilia, "Pengembangan Prototipe *Semantic* Portal Berbasis Ontologi Komunitas Riset: Studi Kasus Fakultas Ilmu Komputer Universitas Indonesia," B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2008.