

# APLIKASI PENCARIAN BUKU PERPUSTAKAAN DENGAN MENGGUNAKAN RDF DAN SPARQL

## ABSTRAK

Dengan berkembangnya jumlah informasi yang bervariasi dan tersebar di berbagai tempat, diperlukan adanya suatu teknologi yang dapat menyatukan informasi tersebut dan menyajikannya sesuai dengan konteks yang diinginkan pengguna. Salah satu teknologi yang dapat mengatasi hal tersebut adalah teknologi Web Semantik. Penelitian ini dilakukan untuk membuat aplikasi pencarian buku menggunakan pendekatan Web Semantic. Aplikasi pencarian ini ditujukan untuk mempermudah pengguna dalam melakukan pencarian buku. Pada aplikasi ini, struktur pengetahuan buku perpustakaan dirancang dengan menggunakan RDF. Sedangkan untuk perancangan query digunakan SPARQL dan untuk menjembatani RDF dan PHP digunakan RAP.

*Kata kunci: Web Semantik, Perpustakaan, RDF, RAP, SPARQL*

Javier Zebua  
Metty Mustikasari

Sistem Informasi  
Fakultas Ilmu Komputer dan teknologi Informasi  
Universitas Gunadarma  
javierbua@yahoo.com  
netty@staff.gunadarma.ac.id

## PENDAHULUAN

Kebutuhan masyarakat akan informasi saat ini merupakan kebutuhan yang sangat mendasar untuk menentukan rencana kedepan. Dengan akses internet yang sudah tidak terbatas saat ini, hampir semua keingintahuan masyarakat akan informasi yang dibutuhkannya dapat terjawab. Untuk dapat memperoleh informasi apa saja yang sesungguhnya benar-benar dibutuhkan, maka pengguna harus dapat memilah-milah website mana saja yang akan diakses. Beberapa website telah muncul untuk memberikan fasilitas layanan pencarian data yang dimaksud. Namun, deretan data yang ditampilkan tidak terstruktur sehingga pencarian berjalan tidak efektif karena memakan waktu yang cukup lama untuk membuka setiap halaman website.

Untuk menangani permasalahan tersebut, dikembangkan website berbasis Semantik. Website berbasis semantik ini akan mampu memahami makna dari sebuah kata atau konsep dan mampu memahami hubungan logis di antara keduanya. Sehingga web semantik hanya akan menampilkan data dan informasi yang dibutuhkan oleh pengunjung website tersebut.

Pada penelitian ini dibuat sebuah aplikasi pencarian yang menggunakan teknologi semantik web yang dipergunakan untuk pencarian buku perpustakaan pada SMK Yadika 7 Bogor.

## TINJAUAN PUSTAKA

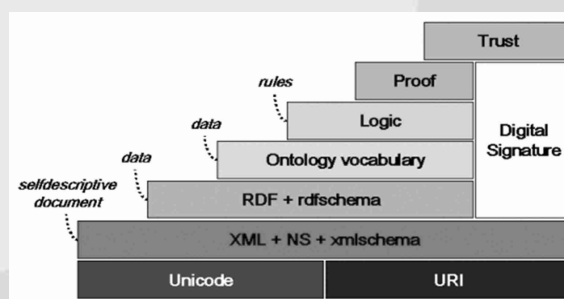
### Web Semantik

Secara umum web semantik dapat didefinisikan sebagai Perluasan Web, dimana informasi yang tersedia lebih bermakna, sehingga manusia dapat berinteraksi dengan komputer dengan lebih baik [1]. Web Semantik dapat berisi informasi dalam jumlah sangat besar di *world wide web* yang terhubung secara global dengan suatu cara tertentu dan dimengerti oleh mesin, sehingga dapat diproses secara langsung oleh mesin menjadi *knowledge* untuk ditampilkan kepada *user*. Dengan metode *Web Semantic*, data berbasis HTML dapat

diubah menjadi format yang dapat dipahami oleh mesin, sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan antara informasi[1,3].

### Arsitektur Web Semantik

Prinsip-prinsip web semantik diimplementasikan dalam lapisan teknologi web dan standar. Gambar berikut ini akan menyajikan lapisan teknologi yang dibutuhkan dalam arsitektur web semantik[1].



Gambar 1. Arsitektur web semantic

Lapisan *unicodedan* URI memastikan digunakannya set karakter internasional dan menyediakan sarana untuk mengidentifikasi objek dalam web semantik. Lapisan XML dengan definisi skema *namespace* memastikan untuk mengintegrasikan definisi web semantik dengan standar berbasis XML lainnya. Dengan RDF dan RDFS akan memungkinkan untuk membuat pernyataan tentang objek dengan URI dan menentukan apakah kosakata tersebut dapat disebut dengan URI. Lapisan ini merupakan lapisan dimana diberikannya jenis sumber daya dan link. Lapisan Ontologi mendukung evolusi kosakata karena dapat menentukan hubungan antara konsep-konsep yang berbeda. Lapisan *Digital Signature* digunakan untuk mendeteksi perubahan pada dokumen.

### Ontology

Dalam dunia komputer ontologi digunakan untuk menspesifikasikan suatu konseptualisasi [3,5]. Dalam istilah lain ontologi dijelaskan sebagai suatu representasi dari domain pengetahuan

tertentu yang berisi istilah-istilah dalam domain tersebut beserta hubungan antara istilah-istilah yang ada.

Saat ini Ontologi banyak digunakan terutama untuk mendukung web semantik yang diarahkan untuk dapat memahami makna suatu kata atau kalimat yang diberikan oleh pengguna. Membuat komputer mengerti seperti manusia bukan merupakan suatu hal yang mudah, namun hal ini terus diupayakan dengan menyediakan seperangkat alat sehingga membuat mesin atau komputer dengan mudah dapat memproses informasi dan mengerti informasi yang diinginkan oleh pengguna[3,5].

Agar dapat digunakan, ontologi sendiri mempunyai struktur bahasa yang formal atau terdefinisi. Struktur bahasa yang dapat menyusun *ontology* antara lain adalah : XML (*Extensible Markup Language*), XML Schema, RDF (*Resource Description Framework*), RDF Schema, OWL (*Ontology Web Language*) [5].

### RDF (Resource Description Framework)

RDF merupakan suatu metadata yang digunakan untuk mendeskripsikan alamat sumber daya pada web[7,8]. RDF memiliki bagian-bagian standar yang digunakan untuk membentuknya antara lain *resource* yang digunakan untuk menggambarkan apa saja yang dapat dimiliki oleh sebuah URI, *Property* berisikan nama *property-property* yang ada dalam *resource, property value* yang berisikan nilai dari sebuah *property*. Dari bagian-bagian tersebut nantinya dapat membentuk sebuah *statement* yang berisikan *resource, property* dan *property value* atau yang di kenal sebagai *triple* (3-tuple) yang berfungsi sebagai penyimpanan data dan relasi antar data.

RAP (RDF API for PHP) merupakan tools web semantik untuk bahasa PHP yang memiliki fitur untuk memanipulasi, menyimpan, melakukan *query* serta membangun graph RDF[8]. Inti dari RAP meliputi dua implementasi atas penyimpanan *statement*, yaitu penyimpanan *graph* RDF dalam *memory*

sistem maupun dalam *database relational*.

### SPARQL

Model data RDF berupa suatu statemen dalam bentuk triple yang terdiri dari subjek, predikat, dan objek[8]. Untuk mendapatkan informasi dari suatu graph RDF dibutuhkan suatu *query*. SPARQL merupakan suatu bahasa *query* yang dapat digunakan untuk mengakses data pada web semantic[3]. Dengan menggunakan SPARQL memungkinkan untuk : (1) Mengambil nilai dari data yang terstruktur maupun data yang semi terstruktur; (2) Mengembangkan data dengan melakukan *query* terhadap suatu relasi yang tidak diketahui; (3) Dapat melakukan *query* operasi *join* yang kompleks pada *database* yang berlainan secara lebih sederhana; (4) Mengubah suatu data RDF menjadi vocabulary yang lain. Hasil dari *query* SPARQL dapat mengembalikan nilai dalam beberapa format data yang antara lain : XML, RDF, dan HTML.

### METODE PENELITIAN

Pada penelitian ini, langkah pertama yang dilakukan adalah melakukan telaah pustaka yang berkaitan dengan Web Semantik, XML, OWL, RDF, RAP, PHP dan SPARQL. Kemudian langkah kedua dilanjutkan dengan pembuatan *Resource Description Framework* (RDF). Dari data RDF yang telah dibuat akan dilanjutkan dengan perancangan *query* untuk dapat menghasilkan *output* yang diinginkan dengan menggunakan SPARQL. Langkah selanjutnya adalah pembuatan tampilan aplikasi dengan menggunakan PHP dan untuk menjembatani RDF dan PHP digunakan RAP. Langkah terakhir adalah implementasi program aplikasi dan melakukan ujicoba program aplikasi yang telah dibuat.

### HASIL DAN PEMBAHASAN

#### Perancangan

Perancangan dimulai dengan perancangan struktur pengetahuan dengan menggunakan RDF, perancangan query dengan SPARQL dan perancangan tampilan aplikasi. Perancangan RDF dilakukan untuk menyimpan informasi dari buku-buku yang ada di perpustakaan. RDF tersebut akan menyimpan informasi dari setiap buku berupa judul buku, penulis buku, penerbit buku, kategori buku, tahun terbit, jumlah buku, klasifikasi buku, asal buku, nomor induk buku, rak buku dan keterangan buku. Bentuk RDF yang dibuat akan terlihat seperti berikut ini :

```
<rdf:RDF
xmlns:Description="http://localhost/javier/buku.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ns1="http://www.w3.org/2001/vcard-rdf/3.0#"
xmlns:owl="http://www.w3.org/2002/07/owl#">
```

```
<rdf:Description
rdf:about="http://javierzebuga.com/ontologies/buku.owl#buku_agama1">
<ns1:isA>Buku</ns1:isA>
<ns1:hasJudul>2T Taklukan Takdirmu
!</ns1:hasJudul>
<ns1:hasKategori>Agama</ns1:hasKategori>
<ns1:hasPenulis>Rusdin S. Raup</ns1:hasPenulis>
<ns1:hasPenerbit>Hikmah</ns1:hasPenerbit>
<ns1:hasTahunTerbit>2008</ns1:hasTahunTerbit>
<ns1:hasJumlah>5</ns1:hasJumlah>
<ns1:hasKlasifikasi>297</ns1:hasKlasifikasi>
<ns1:hasAsalBuku>Yadika Pusat</ns1:hasAsalBuku>
<ns1:hasNoInduk>0422/6-IV/2001</ns1:hasNoInduk>
<ns1:hasRak>3</ns1:hasRak>
<ns1:hasKeterangan>Yayasan</ns1:hasKeterangan>
</rdf:Description>
</rdf:RDF>
```

RDF di atas menggambarkan bahwa model RDF tersebut memiliki resource dari URI [http://javierzebuga.com/ontologies/buku.owl#buku\\_agama1](http://javierzebuga.com/ontologies/buku.owl#buku_agama1). Di dalam *resource* tersebut terdapat *property-property* seperti *isA*, *hasJudul*, *hasKategori*, *hasPenulis*, *HasPenerbit*, *hasTahunTerbit*, *hasJumlah*, *hasKlasifikasi*, *hasAsalBuku*, *hasNoInduk*, *hasRak* dan *hasKeterangan* yang mempunyai *prefix* yang berasal dari *namespace* *ns1*. *Property* tersebut juga memiliki *property value* sebagai contoh memiliki *property value* sebagai contoh *property value* *Rusdin S. Raup*.

Untuk membentuknya ke dalam sebuah statement RDF (*triple*) maka diperlukan bagian-bagian yang diterjemahkan sebagai subjek, predikat dan objek. Pada RDF tersebut, yang bertindak sebagai subjek adalah *resource*, yang bertindak sebagai predikat adalah *property* dan objeknya merupakan *property value* dari RDF.

Berikut ini merupakan contoh *statement* dari RDF di atas :

```
http://javierzebuga.com/ontologies/bu
```

```
ku.owl#buku_agama1 hasPenulis
Rusdin S. Raup
S P O
```

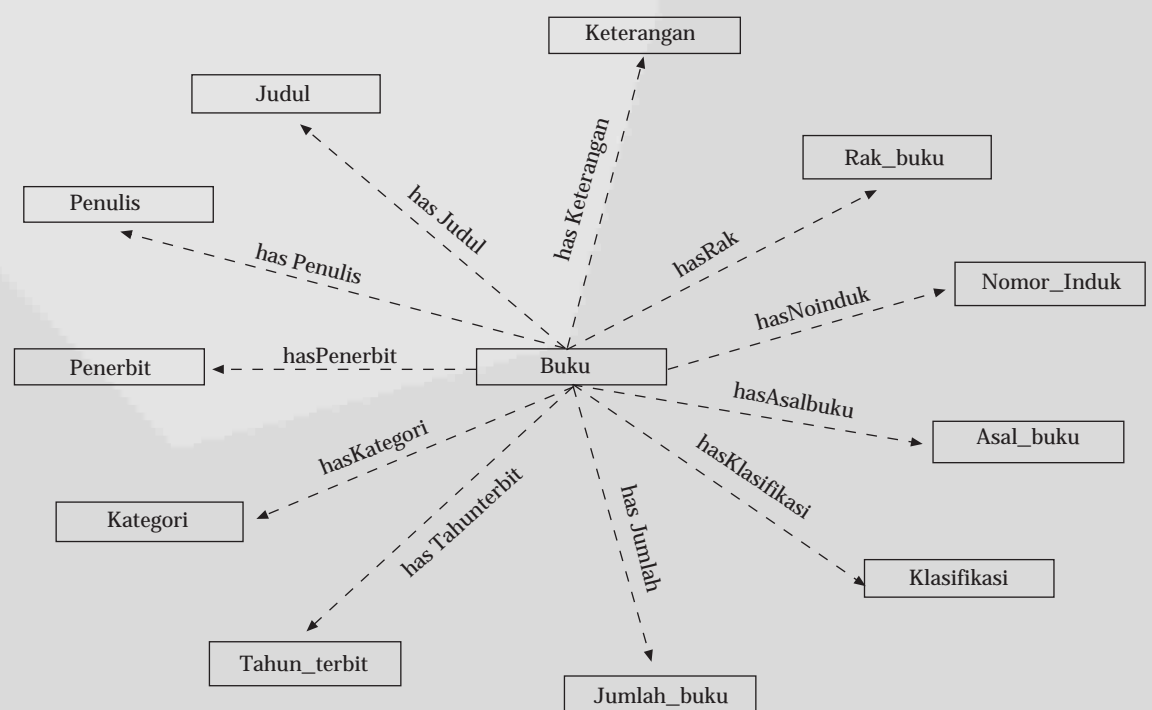
Setiap *resource* dari RDF tersebut memiliki dua belas *property* sehingga dapat dibentuk dua belas *statement* dimana *resource* tersebut menjadi subjek, predikat diambil dari *property* yang tersedia serta objek diambil dari *property value* yang ada. Dengan terbentuknya *statement* maka penggambaran visual dari sebuah RDF atau yang disebut dengan *graph* dapat terlihat. Gambar berikut ini akan memberikan penggambaran tentang struktur *graph* dari RDF di atas :

Setelah RDF yang digunakan untuk menyimpan informasi dari setiap buku terbentuk, maka untuk mengambil informasi dari RDF tersebut diperlukan *query* SPARQL. Untuk penerapannya pada aplikasi pencarian ini, digunakan *query* SPARQL *select* untuk menampilkan informasi tersebut di atas. Di bawah ini adalah salah satu *query* SPARQL yang digunakan :

```
select ?judul ?penulis ?penerbit ?kategori
?tahunTerbit ?jumlah ?klasifikasi
?asalBuku ?noInduk ?rak ?keterangan
WHERE {
```

```
?x vcard:hasJudul ?judul.
?x vcard:hasPenulis ?penulis.
?x vcard:hasPenerbit ?penerbit .
?x vcard:hasKategori ?kategori .
?x vcard:hasTahunTerbit ?tahunTerbit .
?x vcard:hasJumlah ?jumlah.
?x vcard:hasKlasifikasi ?klasifikasi .
?x vcard:hasAsalBuku ?asalBuku .
?x vcard:hasNoInduk ?noInduk .
?x vcard:hasRak ?rak .
?x vcard:hasKeterangan ?keterangan .
FILTER regex (?judul, "kata_kunci", "i") // regex
(?penulis, "kata_kunci", "i") // regex (?penerbit,
"kata_kunci", "i") // regex (?kategori,
"kata_kunci", "i");
```

Untuk *query* di atas, digunakan untuk menampilkan informasi tentang judul buku, penulis buku, penerbit buku, kategori buku, tahun terbit buku, jumlah buku, klasifikasi buku, asal buku, nomor induk buku, rak buku dan keterangan



Gambar 2. Struktur Graph RDF Buku



buku. Untuk *query* tersebut, kata kunci yang diambil dapat berupa salah satu dari judul, penulis, penerbit atau kategori buku.

Setelah pembuatan RDF dan *query-query*, maka tahap selanjutnya adalah perancangan tampilan dari aplikasi yang akan dibuat. Pada program aplikasi tersebut digunakan RAP (RDF API for PHP) sebagai *framework* yang berfungsi untuk mengkonversi RDF yang ada menjadi informasi yang dapat diterima oleh PHP. Sedangkan untuk SPARQL disisipkan sebagai *query* pada pemograman PHP di setiap halaman yang membutuhkan *query-query* untuk mengambil informasi dari RDF.

### Perancangan Tampilan Aplikasi

Untuk perancangan tampilan aplikasi, dimulai pada rancangan tampilan halaman pencarian seperti terlihat pada gambar berikut :

BANNER					
Home		Category		About	Contact
Kata kunci : <input type="text"/> <input type="button" value="cari"/>					
Pencairan dengan kata kunci ? kata_kunci, ditemukan x hasil					
No.	Judul	Penulis	Penerbit	Kategori	Detail
					[lihat]
					[lihat]
					[lihat]
Copyright © 2011 smk-yadika-7-bogor. All Right Reserved					

Gambar 3. Rancangan tampilan halaman pencarian

Rancangan di atas merupakan rancangan aplikasi pada halaman awal atau halaman pencarian. Dalam rancangan tersebut disediakan form pencarian dengan meng-*input* kata kunci dan hasil pencarian juga akan disajikan dalam bentuk tabel. Untuk rancangan tampilan pada halaman pencarian berdasarkan kategori terlihat pada gambar berikut :

BANNER					
Home		Category		About	Contact
CATEGORY .:Gunakan filter yang tersedia					
Kata Kunci : <input type="text"/>					
Kategori Buku : <input type="text"/>					
Cari Berdasarkan : <input type="text"/>					
<input type="button" value="Cari"/>					
Pencairan dengan kata kunci ? kategori buku? dan berdasarkan ? di temukan x hasil					
No.	Judul	Penulis	Penerbit	Kategori	Detail
					[lihat]
					[lihat]
					[lihat]
Copyright © 2011 smk-yadika-7-bogor. All Right Reserved					

Gambar 4. Rancangan tampilan halaman pencarian berdasarkan kategori

Pada rancangan tampilan halaman pencarian berdasarkan kategori, pengguna akan diminta untuk menginput kata kunci, memilih kategori buku, dan melakukan pencarian berdasarkan judul buku, penulis buku atau penerbit buku. Untuk rancangan tampilan detail buku dapat dilihat pada gambar berikut :

BANNER			
Home	Category		About
Informasi Buku			
Judul Buku	:		
Penulis	:		
Penerbit	:		
Kategori	:		
Tahun terbit	:		
Jumlah	:		
Klasifikasi	:		
Asal Buku	:		
Nomor Induk	:		
Rak	:		
Keterangan	:		
Copyright © 2011 smk-yadika-7-bogor. All Right Reserved			

Gambar 5. Rancangan tampilan halaman detail buku

Rancangan di atas dibuat untuk dapat menampilkan secara detail informasi yang dimiliki oleh sebuah buku. Untuk rancangan tampilan pada halaman *about* dapat dilihat pada gambar berikut :

Untuk halaman *about* dirancang hanya untuk memberikan penjelasan singkat tentang web pencarian dan menampilkan koleksi kategori buku yang

atau navigasi tiap halaman pada suatu situs website. Pada perancangan website ini, terdapat satu struktur navigasi yang digunakan yaitu struktur navigasi *user*. Struktur navigasi yang diterapkan adalah struktur navigasi campuran. Berikut ini gambarannya :

Dalam pembuatan program aplikasi digunakan beberapa perangkat lunak pendukung seperti berikut yaitu RAP Library, Xampplite, PHP, Sistem operasi Windows XP SP3, Google chrome sebagai web browser.

### UJI COBA

Uji coba dilakukan untuk mengetahui apakah pencarian buku pada aplikasi ini sudah sesuai dengan yang diharapkan. Uji coba dilakukan pada web pencarian buku berbasis semantik yang sudah dibuat.

BANNER			
Home	Category		Contact
Sekilas tentang website			
Koleksi			
Copyright © 2011 smk-yadika-7-bogor. All Right Reserved			

Gambar 6. Rancangan tampilan halaman *about*

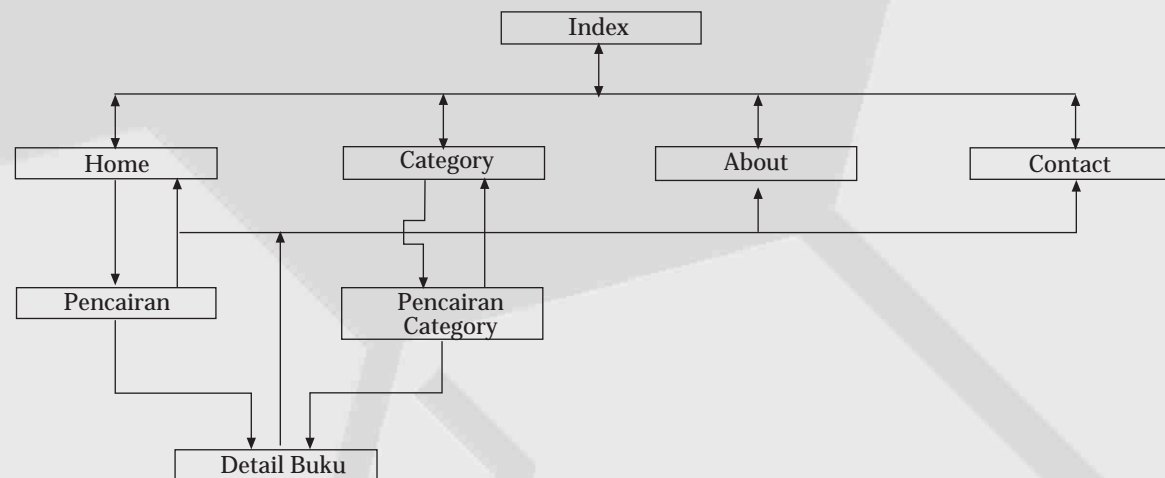
Pengujian dilakukan pada halaman pencarian dan pada halaman pencarian *category*. Pada halaman pencarian *category* pengujian dilakukan dengan cara memasukkan kata kunci dan melakukan pemilihan pada dua kategori pilihan yaitu memilih kategori buku dan pemilihan dasar pencarian.

Uji coba pada halaman pencarian dilakukan dengan memasukkan kata kunci ke dalam *form* yang tersedia. Kata kunci yang dimasukkan akan dijadikan sebagai *keyword* yang digunakan oleh aplikasi dalam menjalankan proses pencarian berbasis RDF. Aplikasi ini akan mencari kecocokan antara kata kunci yang ada dengan kata yang dikandung oleh bagian judul buku, penulis, penerbit atau kategori buku.

Gambar di atas menunjukkan hasil pencarian yang telah dilakukan. Saat melakukan uji coba, kata kunci yang dimasukkan adalah "hati". Dengan penggunaan kata kunci tersebut, ditemukan hasil pencarian data sebanyak 7.



Gambar 7. Rancangan tampilan halaman *contact*



Gambar 8. Struktur navigasi campuran



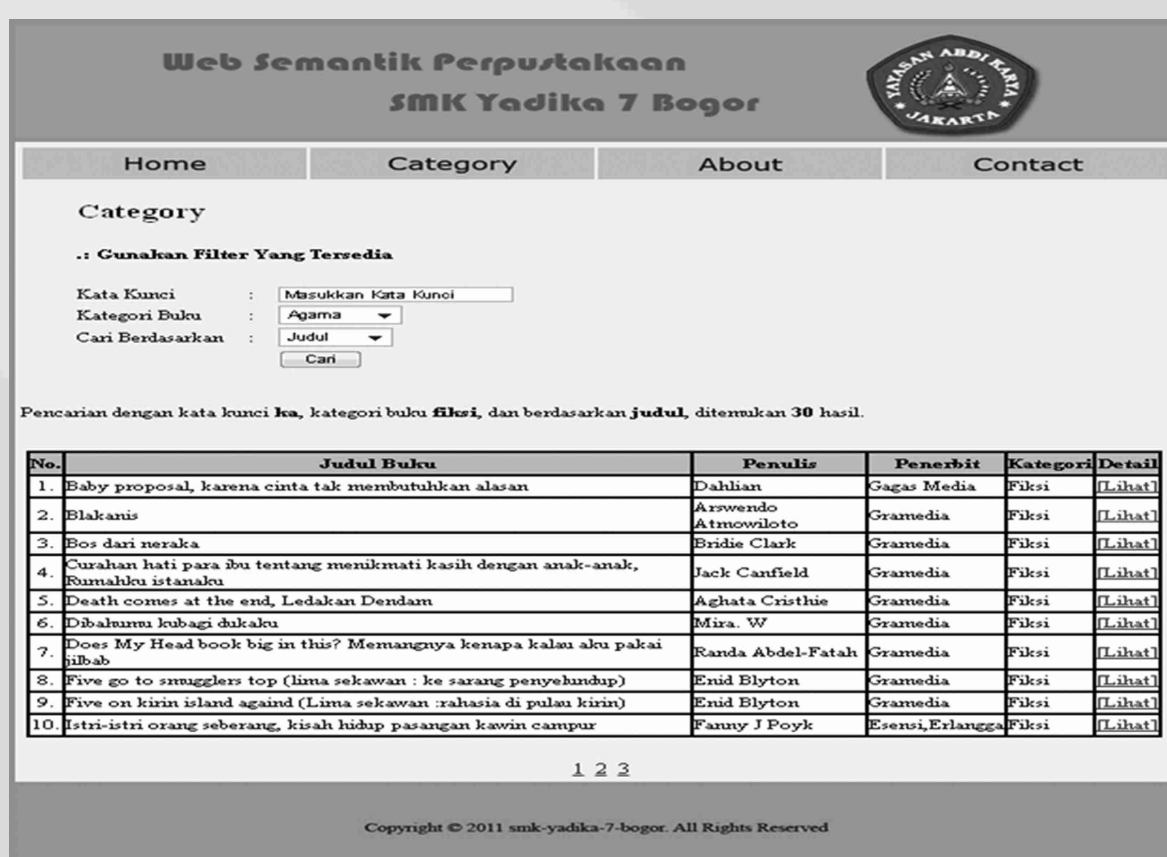
Gambar 9. Pencarian dengan kata kunci hati

Pada halaman pencarian, kategori pengujian dilakukan dengan cara memasukkan kata kunci, memilih kategori buku dan melakukan pencarian buku berdasarkan kategori yang ada (judul, penulis, penerbit). Pada pengujian ini dilakukan dengan memasukkan kata kunci *ka*, kategori buku *fiksi*, dan pencarian berdasarkan *judul*. Hasil pencarian yang dilakukan berdasarkan kata kunci dan kategori-kategori tersebut didapatkan 30 hasil. Berikut adalah tampilan hasil pencariannya :

### Analisis

Penggunaan metode semantik yang salah satunya menggunakan teknologi ontology memiliki struktur atau hirarki dari sebuah domain yang dapat digunakan untuk mencari sumber informasi yang relevan dengan data yang diinginkan.

Dalam pelaksanaan uji coba di atas, data perpustakaan dijadikan sebagai masukan dari aplikasi ini. Berbagai Kata Kunci dapat digunakan dalam program



Gambar 10. Pencarian dengan kata kunci ka, kategori buku fiksi dan berdasarkan judul

aplikasi pencarian ini seperti judul buku, nama penulis dan penerbit. Adapun tingkat keberhasilan dari Web tersebut dilihat dari ketepatan dalam pencarian data yang diperlukan.

Dari hasil uji coba yang dilakukan, dengan melihat hasil pencarian menggunakan berbagai jenis kata kunci, hasil temu kenali yang dihasilkan sudah relevan dengan kata kunci yang dimasukkan. Seluruh data yang terkait dengan dengan kata kunci muncul dengan berbagai variasi data sesuai dengan data yang tersedia pada database. Program aplikasi ini juga dievaluasi oleh 20 pengguna. Dari hasil evaluasi pengguna, diperoleh bahwa program pencarian sudah cukup baik dan relevan dengan yang diinginkan pengguna. Tetapi informasi yang tersedia masih kurang, sehingga data yang ada perlu ditambah.

### PENUTUP

Pada aplikasi ini struktur pengetahuan yang digunakan disajikan dalam bentuk RDF dan *query* yang digunakan untuk mendapat informasi dari RDF tersebut disajikan dengan menggunakan SPARQL. Pencarian dapat dilakukan dengan beberapa kata kunci seperti judul buku, nama penulis dan penerbit. Pada aplikasi ini setiap fitur pencarian yang tersedia di uji apakah relevan dengan hasil yang diharapkan. Dari hasil uji coba, aplikasi ini telah mencapai tujuan yang diharapkan, semua data yang ditemui kenali pada proses uji coba relevan dengan kata kunci yang dimasukkan dan sesuai dengan kebutuhan pengguna. Tetapi informasi yang tersedia masih kurang, sehingga perlu adanya penambahan data.

M-R. Koivunen, E. Miller (2001). W3C Semantic Web Activity. Proceedings of the Semantic Web Kick-off Seminar in Finland. <http://www.w3.org/2001/12/semweb-fin/w3csw0210/2004>



J.H.T Bekke (2000),. Semantic Data Modelling, Prentice Hall, Hemel, Hemstead.

Edd Dumbill. (2000). The Semantic Web: A Primer.

[http://www.xml.com/pub/a/2000/11/01/semantic\\_web/index.html](http://www.xml.com/pub/a/2000/11/01/semantic_web/index.html)

Matthews, Brian, (2005) Semantic Web Technologies, JISC Technology and Standards Watch Semantic Web Technologies.

[http://www.jisc.ac.uk/uploaded\\_documents/jisctsw\\_05\\_02bpdf.pdf](http://www.jisc.ac.uk/uploaded_documents/jisctsw_05_02bpdf.pdf)

Wicaksana, I Wayan Simri. (2006). Ontology: Bahasa dan Tools Protégé. Universitas Gunadarma, Jakarta.

Provoost, Lee & Bornier, Erwan. (2006). Service-Oriented Architecture and the Semantic Web: A killer combination? Published Thesis. University of Utrecht, Netherland.

Pierre Antonine Champin. ( 2001). RDF Tutorial.  
<http://WWW.W3.org/2000/02/rdf-tutorial.pdf>

W3C Recommendation. Agustus 2011. RDF: Concept and Abstract Syntax.  
<http://www.W3.org/TR/2004/REC-rdf-concept-2004>.

