

BAB 4

PEMODELAN DAN IMPLEMENTASI SISTEM

4.1. Analisis Dan Perancangan Sistem

4.1.1. Analisis Masalah

Permasalahan utama yang diangkat dalam penelitian ini adalah membentuk dan menyediakan data bibliografi dalam bentuk *linked data* pada web. Pada umumnya banyak data di simpan kedalam bentuk relational database termasuk data tentang bibliografi. Kebanyakan data bibliografi direpresentasikan hanya dalam bentuk dokumen (HTML) yang didalamnya terdapat *hyper-link* untuk merujuk ke sebuah tautan web tertentu. Sehingga mesin-mesin lain yang akan menggunakan data tersebut sulit untuk menggambarkan sebuah pengetahuan pada dokumen termasuk pada *hyper-link* tersebut. *Linked data* bibliografi tidak hanya menggambarkan sebuah pengetahuan dari sebuah halaman di satu mesin, tetapi juga pengetahuan dari mesin-mesin eksternal lain yang dibentuk dalam sebuah halaman sesuai data-data terkait yang dibutuhkan untuk menyajikan sebuah informasi bibliografi.

Dalam mempublish data bibliografi sebagai *linked data* terdapat masalah yang cukup kompleks karena harus memperhatikan beberapa faktor, seperti data awal yang cukup kompleks karena harus memperhatikan data awal tentang bibliografi, membentuk setiap data dalam tabel-tabel database bibliografi kedalam bentuk RDF dengan proses mapping dan tetap memperhatikan hubungan atau relasi antar table didalamnya, membentuk resource dari setiap data dengan mengacu pada prinsip *linked data* yang bukan saja dapat diakses dari mesin internal tapi juga dapat diakses mesin-mesin lain serta dapat mengambil resource dari sumber-sumber data mesin eksternal lain dengan jalan mengidentifikasi setiap resource sebagai URI dan mengaitkan resource satu dengan yang lainnya agar terbentuk data bibliografi yang menggabungkan data dari beberapa mesin atau sumber data lain dalam teknologi *linked data*. Terdapat beberapa data yang diambil dari sumber lain, data-

data yang tersimpan dari beberapa sumber diakses langsung sehingga data tersebut tidak disimpan kembali dalam sistem bibliografi.

Data yang diambil dari sumber data lain seperti Dbpedia tentang informasi seorang penulis atau tokoh dari sebuah karya tulis karena secara khusus *linked data* bibliografi tidak menyediakan data atau informasi tambahan mengenai penulis atau tokoh yang dimaksud. Hasil dari mempublish data bibliografi sebagai *linked data* dapat ditampilkan dengan melakukan query oleh sistem lokal, dan dapat juga ditampilkan oleh mesin-mesin lain dalam teknologi *linked data*, dengan kata lain mesin-mesin lain juga dapat menggunakan web of data tersebut atau mengaitkan datanya dengan data bibliografi yang telah dibangun.

4.1.2. Ruang Lingkup Sistem

Penelitian ini bertujuan untuk membangun sebuah repository bibliografi berbasis *linked data* dengan memanfaatkan beberapa sumberdaya dan aplikasi yang telah ada dengan beberapa fasilitas yaitu :

1. Menyediakan data bibliografi dalam bentuk *linked data*
2. Menyediakan fasilitas edit file mapping database ke bentuk bentuk RDF
3. Menampilkan query sparql yang disediakan dalam bentuk SPARQL endpoint
4. Menyediakan fasilitas untuk menampilkan informasi bibliografi dalam bentuk HTML dengan mengaitkan sumberdaya eksternal lainnya.

4.1.3. Aturan Bisnis

Di dalam proses pengembangan sebuah sistem perlu ditentukan aturan bisnis untuk memberikan batasan-batasan apa yang dilakukan sistem maupun apa yang tidak dapat dilakukan oleh sistem yaitu :

1. Sistem mentranslasi database (RDB) yang sudah tersedia ke dalam bentuk *linked data* dengan mengintegrasikan sumber data lain berupa URI yaitu Dbpedia, Worldcat, VIAF.
2. Dalam mentranslasi database disediakan file mapping yang dapat di konfigurasi sewaktu-waktu.

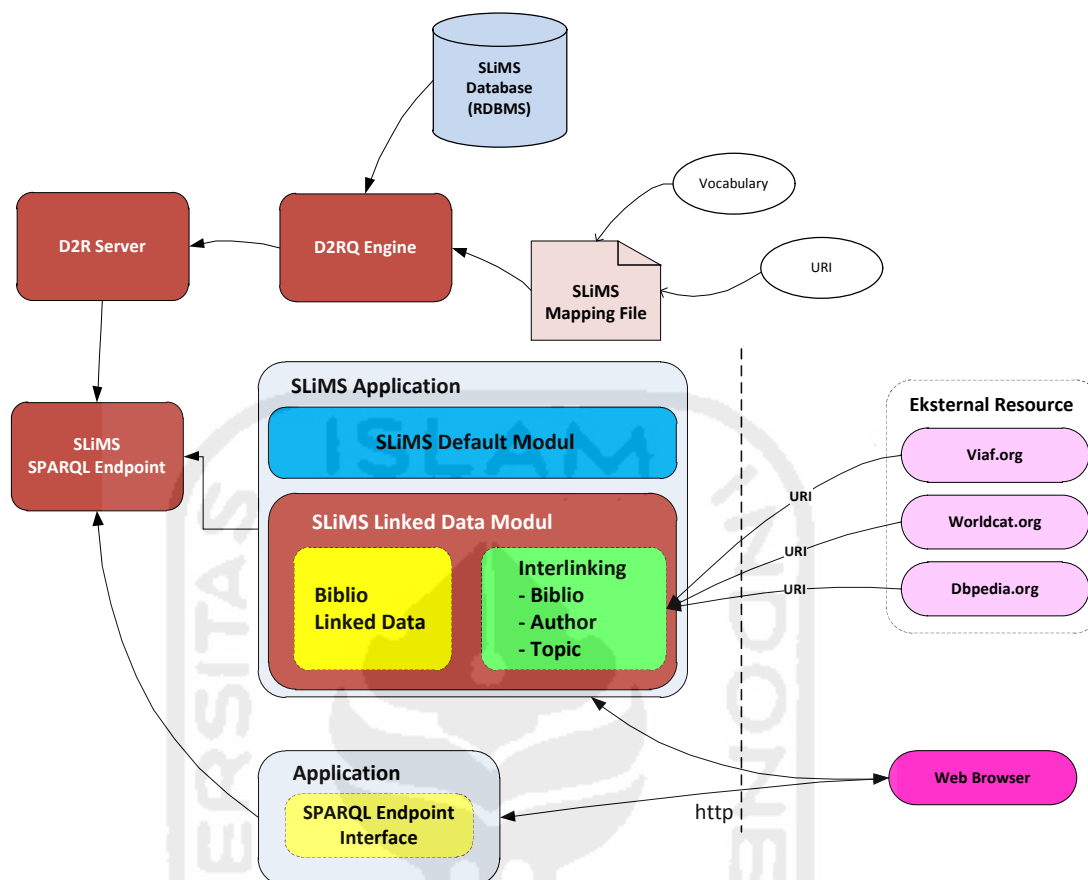
3. Sistem menyediakan *linked data* tentang bibliografi, berupa data koleksi pustaka, author, penerbit, subjek dalam teknologi *linked data* dalam bentuk RDF N-triple.
4. Sistem menyediakan interface SPARQL endpoint untuk melakukan query langsung terhadap data-data yang tersedia.

Sistem menyajikan representasi data dari *linked data* bibliografi dalam aplikasi web, menampilkan tentang data/informasi author, subject/topic, lokasi atau koleksi pustaka yang diambil dari penyedia data eksternal dalam teknologi *linked data* yaitu Dbpedia, Worldcat dan VIAF.

4.1.4. Perancangan Arsitektur Sistem

Arsitektur sistem *linked data* bibliografi terdiri dari bagian-bagian umum yang menggabungkan beberapa komponen-komponen pembentuk sistem *linked data*. Komponen-komponen tersebut terdiri dari sistem internal ataupun sistem yang lainnya dalam bentuk integrasi data.

Pada gambar 4.1 menunjukkan arsitektur sistem, dari gambar tersebut dapat dilihat alur mekanisme kerja sistem. Data bibliografi diambil dari database (RDB/ Non RDF), untuk mengubah menjadi bentuk *linked data* diperlukan adanya mapping yang berfungsi untuk memetakan setiap table yang ada database relasional. File konfigurasi mapping ini disimpan dalam bentuk format file .n3 (N-Triple) yang sewaktu-waktu bisa diedit atau dikonfigurasi ulang melalui aplikasi web. Untuk membuat file mapping diperlukan beberapa data tambahan yaitu vocabulary untuk mendeskripsikan setiap class maupun property dan *linked data* URI yang merupakan URI dari sumber data lain seperti Dbpedia, Worldcat dan VIAF.



Gambar 4.1 Arsitektur Pengembangan Sistem SLiMS Linked Data

File mapping tersebut merupakan rule yang akan digunakan untuk mengkonversi data menjadi bentuk RDF N-Triple yang kemudian bisa diakses dengan menggunakan query langsung melalui SPARQL endpoint, ataupun didownload langsung melalui web. Sistem juga menyediakan representasi informasi data bibliografi dalam bentuk HTML yang dapat dilihat dan dibaca langsung melalui web. Pembentukan *linked data* bibliografi diperlukan beberapa data pendukung dari beberapa sumber yaitu dari Worldcat.org untuk mengetahui informasi bibliografi dan DBpedia.org untuk mengetahui informasi author dan topic dari sebuah bibliografi dengan menyimpan URI pada setiap data yang berkaitan atau mendukung informasi. Dalam pembentukan *linked data* yang didukung oleh *resource eksternal* penulis membuat modul-modul tambahan di dalam aplikasi SLiMS. Modul tambahan tersebut berfungsi untuk membuat interlinking antara data yang terdapat di dalam database “slims” dengan data yang berada di eksternal resource.

4.1.5. Analisa Model Bibliografi

Pada tahap awal dalam perencanaan pembuatan sistem *linked data* bibliografi adalah dengan mengidentifikasi data-data tentang bibliografi. Data bibliografi pada awalnya disimpan di dalam database, kemudian dari data awal tersebut dapat ditentukan gambaran umum RDF Graph dari setiap tabel.

4.1.5.1. Identifikasi Basis Data Bibliografi

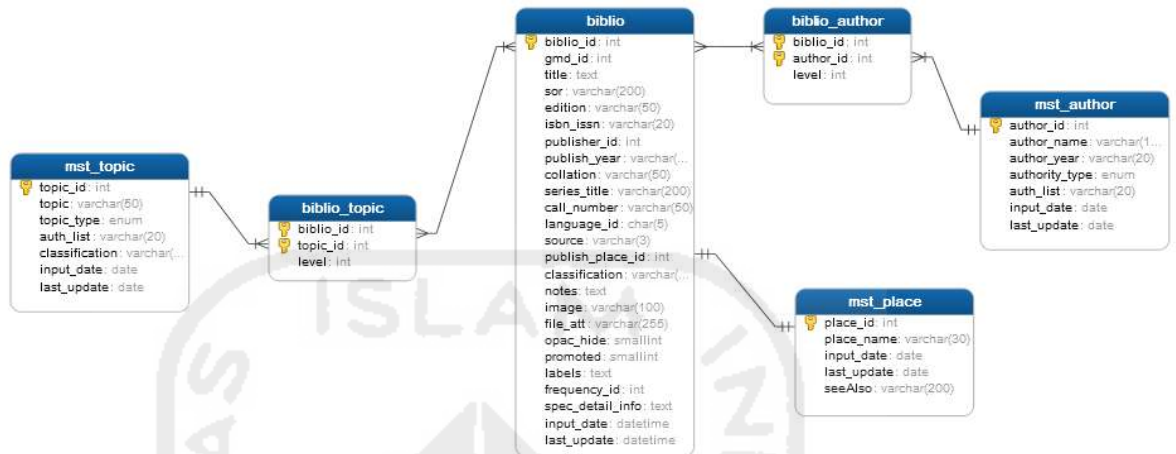
Data bibliografi terbentuk dari data-data yang berisikan data judul dari suatu karya, isbn, author atau penulis dari suatu karya tulis, topic atau subject yang berkaitan dengan hal yang dibahas didalam karya tersebut, penerbit, bentuk fisik dan jumlah halaman.

Setiap data bibliografi mempunyai data judul, author atau penulis dan topic atau subject, dimana informasi data pembentuk bibliografi bisa di per kaya lagi dengan menautkan ke *eksternal resource* lainnya. Informasi author atau penulis bisa di tautkan VIAF. Org (*Virtual International Authority File*) sebagai *eksternal resource*. Topic dan subject dapat di tautkan ke id.dbpedia.org dan dbpedia.org sebagai *eksternal resource*. Semua informasi-informasi tersebut digabungkan untuk membentuk informasi bibliografi dengan teknologi *linked data*.

Untuk menyimpan data-data bibliografi langkah awal dibentuk sebuah database “slims”, database inilah yang akan dipetakan kedalam bentuk struktur RDF (*Resource Description Framework*) sebagai *linked data*. Komponen dari database yang terdiri dari table, coloumns, primary key, foreign key, table relationship dipetakan kedalam bentuk RDF melalui proses mapping kemudian di publish menjadi *linked data* bibliografi.

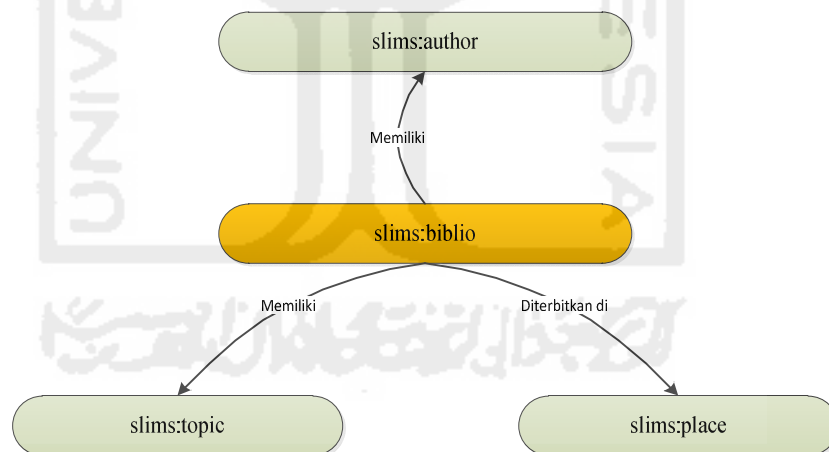
Pada gambar 4.2 menggambarkan skema database “slims”. Pada skema tersebut tidak semua tabel di tampilkan, hanya tabel-tabel yang berkaitan dengan informasi bibliografi yang di gunakan. Pada tabel biblio mempunyai primary key biblio_id, publisher_id sebagai forign key. Pada tabel biblio_topic terdapat dua forign key yaitu biblio_id digunakan untuk relasi dengan tabel biblio dan topic_id sebagai forign_key yang digunakan untuk relasi ke tabel mst_topic. Untuk tabel biblio_author terdapat dua forign key yaitu biblio_id yang berfungsi untuk

membentuk relasi dengan tabel biblio dan author_id yang berfungsi untuk membentuk hubungan dengan tabel mst_author.



Gambar 4.2 Relasi antar tabel aplikasi SLiMS yang terkait bibliografi

Dari skema database “slims” dapat ditransformasikan ke dalam format RDF graph seperti yang terlihat pada gambar 4.3. yang membentuk relasi antar class.



Gambar 4.3 Model RDF Database SLiMS

Dari relasi class yang telah dibentuk, sehingga dapat dilihat hubungan antar class tersebut. Class slims:biblio mempunyai relasi yang berlabel memiliki pada class slims:author, yang dapat diartikan biblio memiliki author tertentu. Class slims:biblio berelasi dengan class slims:topic dengan label memiliki, relasi ini dapat diartikan biblio memiliki topic tertentu. Class slims:biblio juga berelasi dengan class

slims:place dengan label diterbitkan yang memiliki arti bahwa biblio diterbitkan di lokasi tertentu.

4.1.6. Modelling *Linked Data* Bibliografi

Tahap awal untuk mempublikasikan *linked data* bibliografi, terdapat langkah-langkah dengan memperhatikan sejumlah prinsip-prinsip, RDF data dari model sistem *linked data* bibliografi dibuat dengan melakukan pendekatan prinsip-prinsip dasar paradigma *linked data* seperti yang telah dijelaskan pada bab sebelumnya.

4.1.6.1. Desain URI (Uniform Resource Identifier) Bibliografi

Landasan utama pembentukan skema URI adalah untuk mengidentifikasi sumber data pada *linked data* bibliografi. Skema URI terdiri dari beberapa karakter yang mewakili scheme, authority, path, query dan fragment. Desain URI yang akan terbentuk memiliki format sebagai berikut :

1. URI untuk mengidentifikasi data author :

http://opendata-slims.javanehese.net:2021/resource/author/{id_author}

sebagai contoh adalah sebagai berikut :

<http://opendata-slims.javanehese.net:2021/resource/author/10> (author : Fogel, Karl)

scheme : http

authoriry : opendata-slims.javanehese.net:2021

path : author

fragment : 10

2. URI untuk mengidentifikasi data koleksi pustaka :

http://opendata-slims.javanehese.net:2021/resource/biblio/{id_biblio}

Sebagai contoh data adalah sebagai berikut :

<http://opendata-slims.javanehese.net:2021/resource/biblio/11>

(judul : Information Architecture for the World Wide Web: Designing Large-Scale Web Sites)

scheme : http

authority : opendata-slims.javanehese.net:2021

path : biblio

ragment : 11

3. URI untuk mengidentifikasi data topik /subject :

http://opendata-slims.javanehese.net:2021/resource/topic/{id_topic}

Sebagai contoh data adalah sebagai berikut :

<http://opendata-slims.javanehese.net:2021/resource/topic/118> (topic : Novel)

scheme : http

authority : opendata-slims.javanehese.net:2021

path : topic

ragment : 18

4. URI untuk mengidentifikasi data place :

http://opendata-slims.javanehese.net:2021/resource/mst_place/{id_place}

Sebagai contoh data adalah sebagai berikut :

http://opendata-slims.javanehese.net:2021/resource/mst_place/2 (place_name : Jakarta)

scheme : http

authority : opendata-slims.javanehese.net:2021

path : mst_place

ragment : 2

Pada prinsipnya URI pada *linked data* dibuat menggunakan HTTP URI yang mengidentifikasi setiap source yang ada, pada kasus *linked data* bibliografi hal pertama yang dilakukan adalah mentransformasikan tabel-tabel beserta relasi antar tabel yang ada didalam database bibliografi ke dalam bentuk RDF (Resource Description Framework) sesuai dengan prinsip *linked data*, dalam bentuk aturan RDF, URI digunakan untuk mengidentifikasi setiap resource.

4.1.6.2. Model RDF untuk class biblio

Pembentukan skema RDF bertujuan untuk memetakan data pada tabel-tabel serta mendefinisikan URI pada setiap isi data yang ada pada table. Untuk setiap entitas data biblio yang tersedia akan dibuat RDF resource yang diidentifikasi

dengan URI. Pada table 4.1 merupakan table data biblio dalam database SLiMS, table tersebut dipakai sebagai acuan pembuatan RDF beserta URI untuk mendeskripsikan data pada table biblio.



Table 4.1 Table biblio

biblio_id	gmd	title	sor	editio	isbn_issn	publisher_ publish_	collation	series_ call_number	langue	source	publish	classification	notes
17515	93	Producing open source software			9780596007591	2483 1990	xx, 279 p. ; 24 cm.	005.1-22 Fog p	en	(Null)	28	005.1 22	
17513	93	Fashion trends			978-1-8478-8294-3	1834 2011	xv, 167 hal. : il. ; 21 cm	746.92 Kim f		(Null)	16	700	
17512	93	Phenomenology			0-7914-0805-1	3777 1991	vi, 147 hal. ; 21 cm	142.7 Lyo p	en	(Null)	20	100	
17511	93	Understanding art education : En			978-0-415-36739-4	395 2010	ix, 171 hal. ; 21 cm	707.12 Und -	en	(Null)	16	700	Buku ini menjelas
17510	93	The screenwriter activist: writing :			978-415-89714-3	395 2013	xi, 186 hal. ; 21 cm	809.23 Bek t	en	(Null)	20	800	Beberapa film yan
17509	93	The differend: phrases in dispute			0-8166-1610-8	5507 2011	xvi, 208 hal. ; 21 cm	190 Lyo d	en	(Null)	340	100	Buku ini menjelas
17508	93	Desain, sejarah, budaya : Sebuah			978-602-8252-37-9	345 2010	xxxvi, 268 hal. ; 21 cm	745.44 Wal d	id	(Null)	5	700	Seni dan desain ac
17507	16	Interpretasi Lagu Night Rhythm K				5415 2014	xii, 75 hlm.: ilus., 27 cm.	MS/Kha/i/2014	id	(Null)	5	MS/	Karya tulis ini men
17506	16	Preferensi Lagu "Kamu" Grup Col				5415 2014	xi, 76 hlm.: ilus., 27 cm.	MS/Rah/p/2014	id	(Null)	5	MS/	Preferensi musik
17505	16	Perancangan Komposisi Musik A				5415 2014	xi, 62 hlm.: ilus., 27 cm.	MS/Kar/p/2014	id	(Null)	5	MS/	Kebudayaan Aceh
17504	38	Ronggeng Amen Sebagai Identita				5415 2014	xv, 129 hlm.: ilus., 27 cm.	ST-PKJ/Mel/r/2014	id	(Null)	5	ST-PKJ/	ulisan ini menguj
17503	16	Dumaya				5415 2014	xiii, 125 hlm.: ilus., 27 cm.	ST-PCT/Wid/d/2014	id	(Null)	5	ST-PCT/	Dumaya merupaki
17502	38	Perancangan Interior Kantor PT S				5422 2014	xii, 72 hlm.: ilus., 27 cm.	DI/Vis/p/2014	id	(Null)	869	DI/	Saat ini perkantora
17501	16	Garap Gending Glendheng Bendr				5415 2014	xiii, 150 hlm.: ilus., 27 cm	KR/Atn/g/2014	id	(Null)	5	KR/	Penggarapan Geni
17500	16	Proses Pembelajaran Ekstrakuriku				5415 2014	ix, 97 hlm.: ilus., 27 cm.	MS/Mic/p/2014	id	(Null)	5	MS/	Mazmur Musik a
17499	16	Komposisi Musik Odak-odak Ras				5415 2014	xv, 132 hlm.: ilus., 27 cm.	MS/Gin/k/2014	id	(Null)	5	MS/	Suku Karo merupa
17498	16	Gaya Musik Group Debu dalam Li				5415 2014	xi, 113 hlm.: ilus., 27 cm.	MS/Pra/g/2014	id	(Null)	5	MS/	Debu merupakan
17497	16	Pranata				5415 2014	xv, 160 hlm.: ilus., 27 cm.	ST-PCT/Vil/p/2014	id	(Null)	5	ST-PCT/	Pranata merupaka
17496	93	Modelling photography handboc			978-602-00-2261-1	92 2012	vi, 133 hal. : il. ; 23 cm	778.924 Ent m	id	(Null)	2	700	Memotret model r
17495	93	Munculnya elit modern Indonesi			978-979-419-355-6	79 2009	368 hal. ; 21 cm	305.520 959 8 Nie m	id	(Null)	2	300	Buku ini memapar
17494	16	Proses Latihan Brass Section pada				5415 2014	xiv, 95 hlm.: ilus., 27 cm.	MS/Dar/p/2014	id	(Null)	5	MS/	Berbagai macam
17493	16	Fotografi Fashion Hijabers				5423 2014	xii, 91 hlm.: ilus., 27 cm.	FT/Lin/f/2014	id	(Null)	5	FT/	Fotografi Fashion
17492	16	Perkembangan Rumah Adat Meli				5415 2014	xviii, 99 hlm.: ilus., 27 cm.	DI/Sup/p/2014	id	(Null)	5	DI/	Kekayaan budaya
17491	16	Representasi Fenomena Waktu Se				5422 2014	xiv, 91 hlm.: ilus., 27 cm.	SL/Bun/r/2014	id	(Null)	5	SL/	Setiap makhluk di

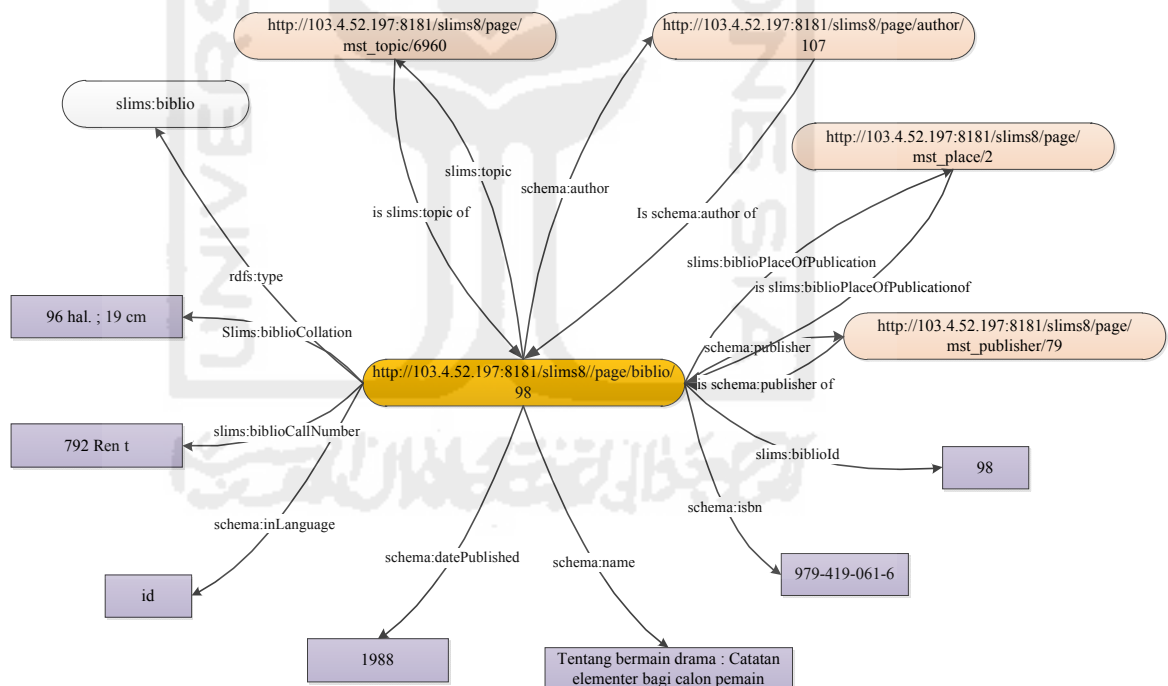
Pada tabel 4.2 mendeskripsikan RDF triple mapping antara tabel biblio dan class biblio. Dari tabel biblio tersebut dibuat sebuah class yaitu slims:biblio. Dari deskripsi setiap entitas dari data biblio dapat di breakdown dengan mengisi atau menggenerate setiap entitas pada data biblio dalam bentuk data RDF Triple dengan menggunakan URI yang sudah dibahas sebelumnya. Dengan menggunakan model D2RQ mapping setiap entitas pada tabel biblio dapat dipetakan menjadi bentuk RDF Triple, terlihat pada predicate schema:publisher merujuk kepada objek publisher_id yang berasal dari tabel mst_publisher dimana biblio.publisher_id = mst_publisher.publisher_id. Pada predikat schema:author sebagai literal merujuk kepada sebuah object author_id yang dihasilkan dari tabel mst_author dimana terdapat relasi tabel biblio_author, biblio dan mst_author yang dimana mst_author.author_id = biblio_author.author_id, biblio_author.biblio_id = biblio.biblio_id.

Table 4.2 RDF Triple Tabel Biblio

Predikat	Objek
rdf:type	slims:biblio
slims:biblioId	biblio.biblio_id
slims:hasTitle	biblio.title
schema:isbn	biblio.isbn_issn
schema:publisher	biblio.publisher_id = mst_publisher.publisher_id
schema:datePublished	biblio.publish_year
slims:biblioCollation	biblio.collation
slims:biblioSeriesTitle	biblio.series_title
slims:biblioCallNumber	biblio.call_number
schema:inLanguage	biblio.language_id
slims:biblioPlaceOfPublication	biblio.publish_place_id
schema:author	mst_author.author_id = biblio_author.author_id, biblio_author.biblio_id = biblio.biblio_id
slims:topic	mst_topic.topic_id = biblio_topic.topic_id, biblio_topic.biblio_id = biblio.biblio_id
slims:biblioPlaceOfPublication	biblio.publish_place_id = mst_place.place_id

Pada gambar 4.4 merupakan graph RDF yang menyajikan informasi biblio dengan alamat URI <http://opendata-slims.javanehese.net:2021/page/biblio/98>, alamat ini merupakan URI yang didalamnya terdapat informasi biblio yang memiliki

kode biblio_id=98, didalam resource tersebut mengakses URI lain yaitu data topic yang terdapat dalam class topic yang mengidentifikasi nama topic tersebut dalam bentuk URI <http://103.4.52.197:8181/slims8/resource/topic/6960>. Kemudian terdapat properti lain yang berisi informasi mengenai author yang ditunjukkan pada properti schema:author yang mempunyai alamat URI http://103.4.52.197:8181/slims8/resource/mst_author/107. Selain itu terdapat beberapa properti lain mengakses URI lain yaitu schema:publisher dengan alamat URI http://103.4.52.197:8181/slims8/resource/mst_publisher/79 yang memuat informasi tentang data penerbit bibliografi yang dimaksud, kemudian yang selanjutnya adalah properti slims:biblioPlaceOfPublication dengan alamat URI http://103.4.52.197:8181/slims8/resource/mst_place/2 yang memuat informasi tentang kota tempat publikasi bibliografi tersebut.



Gambar 4.4 Model Graph RDF Biblio

4.1.6.3. Model RDF untuk class author

Pada tabel 4.3 yang merupakan tabel data author dalam database SLiMS terdapat beberapa kolom yaitu author_id yang merupakan primary key pada tabel

author dalam database dan digunakan sebagai URI yang nantinya akan digunakan oleh class lain sesuai dengan hubungan antar class yang telah dibentuk, kemudian kolom `author_name` yang menyimpan nama author.

Table 4.3 tabel master author

author_id	author_name	author_year	authority_type	auth_list	input_date	last_update	seeAlso
1	RANI, Abd. (ed.)	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
2	SATRANEGARA, Rd. M.	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
3	MAS'ADI, Ghufron A.	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
4	WAHYONO, Effendi (ed.)	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
5	KRISHNAMURTI, J.	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
6	ASHCROFT, Bill	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
7	WIJAYA, Putu	(Null)	p	(Null)	2014-02-18	2014-02-18	http://dbpedia.org/page/Putu_Wijaya
8	BONO, Edward De	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)
9	TEEUW, A.	(Null)	p	(Null)	2014-02-18	2014-02-18	(Null)

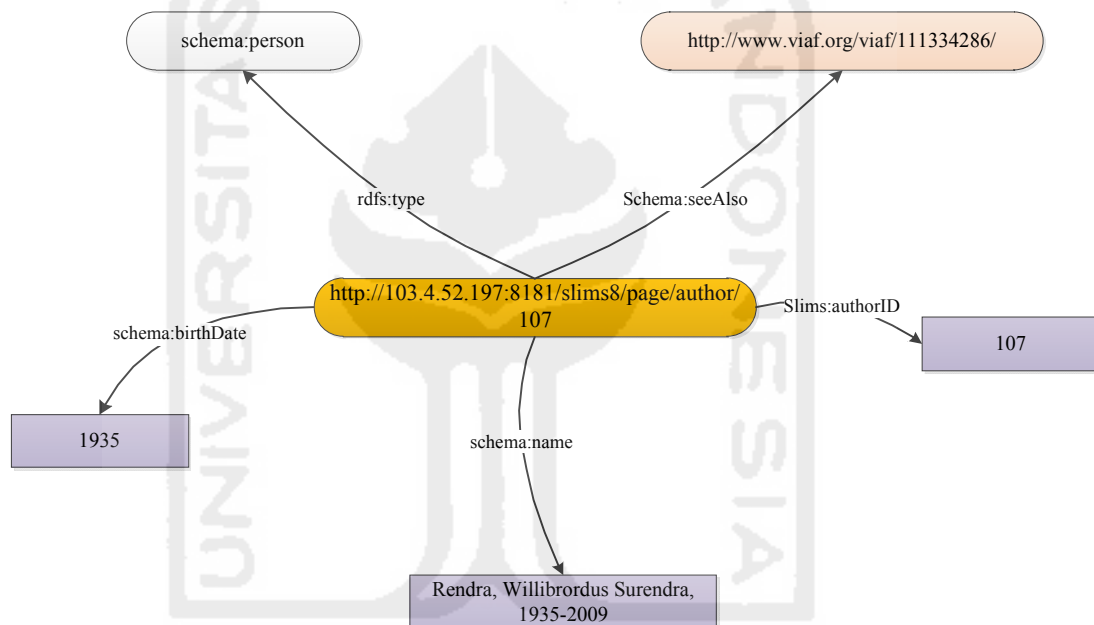
Untuk setiap entitas data author yang tersedia di buat kedalam bentuk RDF, pada tabel 4.4 mendeskripsikan setiap predikat dan object pada RDF untuk setiap entitas data author. Properti `slims:authorId` berisi data dari kolom `mst_author.author_id` pada tabel `mst_author`, properti `schema:name` berisi data dari kolom `mst_author.author_name`, properti `schema:birthDate` berisi data dari kolom `mst_author.author_year`, properti `slims:authorityType` berisi data dari kolom `mst_author.authority_type` dan properti `schema:seeAlso` berisikan data dari kolom `mst_author.seeAlso`.

Table 4.4 RDF triple table author

Predikat	Objek
<code>rdf:type</code>	<code>schema:person</code>
<code>slims:authorId</code>	<code>mst_author.author_id</code>
<code>schema:name</code>	<code>mst_author.author_name</code>
<code>schema:birthDate</code>	<code>mst_author.author_year</code>
<code>slims:authorityType</code>	<code>mst_author.authority_type</code>
<code>schema:seeAlso</code>	<code>mst_author.seeAlso</code>

Seperti yang telah dibahas sebelumnya pada class `biblio` terdapat sebuah URI <http://103.4.52.197:8181/slims8/page/biblio/98> yang merujuk kepada resource lain salah satunya adalah properti `schema:author` dengan alamat URI yaitu <http://103.4.52.197:8181/slims8/resource/author/107> yang dapat diartikan bahwa sebuah data `biblio` dengan kode `biblio_id=98` mempunyai author yang merujuk kepada resource pada class `biblio`, pada Gambar 4.5 merupakan bentuk representasi

dari salah satu data yang ada pada class author yaitu sebuah <http://103.4.52.197:8181/slims8/resource/author/107> yang berasal dari tabel `mst_author` hasil mapping pada class `author`. Predikat `schema:name` merupakan nama author bentuk representatif dari kolom `mst_author.author_name` pada tabel `mst_author`, `schema:birthDate` merupakan tanggal atau tahun kelahiran dari author yang diambil dari kolom `mst_author.author_year` pada tabel `author`. Pada class `author` juga terdapat predikat `schema:seeAlso` dengan objek yang merujuk ke sumber eksternal lain yaitu dengan alamat URI <http://www.viaf.org/viaf/111334286>.



Gambar 4.5 Model Graph RDF Author

4.1.6.4. Model RDF untuk class topic

Untuk class selanjutnya yaitu class `topic`, pada table 4.5 merupakan data pada table `mst_topic` dari database `slims`, tabel tersebut dipakai sebagai acuan pembuatan graph RDF class `topic`. Pada table `mst_topic` tersebut terdapat `topic_id` sebagai primary key yang dipakai sebagai URI, kolom nama `topic` yang menyimpan nama `topic` dan kolom URI yang menyimpan data URI `topic` dari sumber data lain yaitu `dbpedia` yang nantinya informasi tambahan mengenai `topic` tersebut dapat ditampilkan.

Table 4.5 Tabel mst_topic

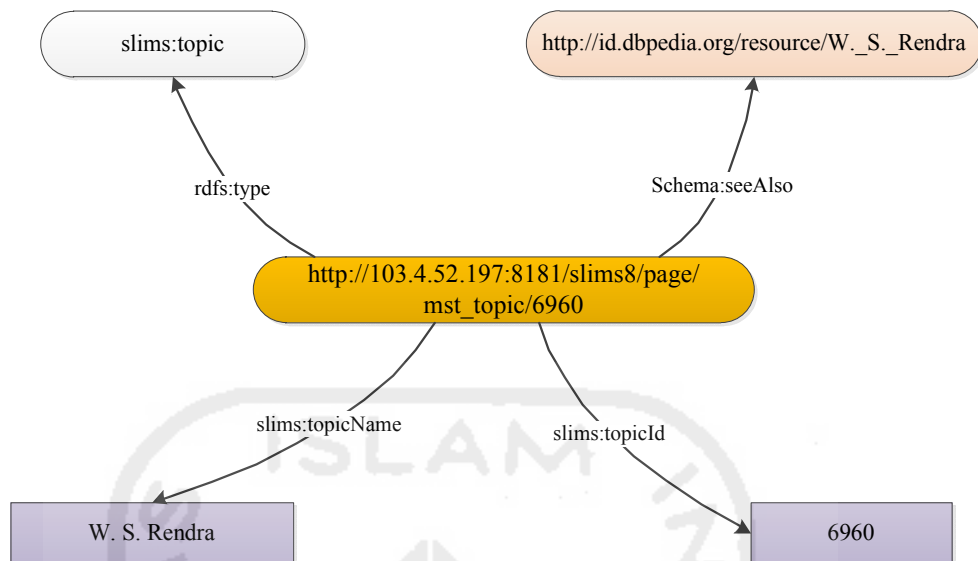
topic_id	topic	topic_type	auth_list	classification	input_date	last_update	see_also
1	Programming	t	(Null)		2007-11-29	2007-11-29	(Null)
2	Website	t			2007-11-29	2015-09-01	http://id.dbpedia.org/resource/Situs_web
3	Operating System	t	(Null)		2007-11-29	2007-11-29	http://dbpedia.org/page/Operating_system
4	Linux	t	(Null)		2007-11-29	2007-11-29	http://id.dbpedia.org/page/Linux
5	Computer	t	(Null)		2007-11-29	2007-11-29	(Null)
6	Database	t	(Null)		2007-11-29	2007-11-29	(Null)
7	RDBMS	t	(Null)		2007-11-29	2007-11-29	(Null)
8	Open Source	t	(Null)		2007-11-29	2007-11-29	(Null)
9	Project	t	(Null)		2007-11-29	2007-11-29	(Null)

Untuk setiap entitas data topic yang tersedia akan di buat RDF yang diidentifikasi dengan URI, pada tabel 4.6 mendeskripsikan setiap predicate dan object pada RDF untuk setiap entitas class topic yang diambil dari data mst_topic. Untuk properti slims:topicName berisi data dari kolom mst_topic.topic dalam tabel mst_topic dan schema:seeAlso yang berisi data dari kolom mst_topic.see_also.

Table 4.6 RDF triple tabel mst_topic

Predikat	Objek
rdf:type	slims:topic
slims:topicId	mst_topic.topic_id
slims:topicName	mst_topic.topic
schema:seeAlso	mst_topic.see_also

Pada gambar 4.6 mendeskripsikan graph RDF class topic. Properti slims:topicName merupakan bentuk representatif dari kolom nama topic pada table mst_topic dan merujuk kembali ke pada sebuah sumber data lain dengan properti schema:seeAlso pada class topic merujuk pada URI http://id.dbpedia.org/resource/W._S._Rendra yang nantinya akan menjelaskan tentang informasi tentang data “W.S. Rendra”, URI tersebut diambil dari *eksternal resource* DBpedia.org.



Gambar 4.6 Model graph RDF topic

4.1.6.5. Model RDF untuk class *mst_place*

Pada tabel yang merupakan tabel data *mst_place* yang menyimpan data lokasi penerbitan sebuah koleksi pustakan. Kolom yang terdapat pada table *mst_place* yaitu *place_id* yang merupakan primary key pada tabel *mst_place* dan digunakan sebagai URI yang nantinya akan digunakan oleh class lain sesuai dengan hubungan antar class yang telah dibentuk, kemudian kolom *place_name* yang menyimpan nama sebuah lokasi.

Table 4.7 Table *mst_place*

place_id	place_name	input_date	last_update	seeAlso
1	Hoboken, NJ	2007-11-29	2007-11-29	(Null)
2	Sebastopol, CA	2007-11-29	2007-11-29	http://dbpedia.org/resource/Sebastopol,_California
3	Indianapolis	2007-11-29	2015-12-19	http://id.dbpedia.org/resource/Indianapolis
4	Upper Saddle River, NJ	2007-11-29	2007-11-29	http://dbpedia.org/resource/Upper_Saddle_River,_New_Jersey
5	Westport, Conn.	2007-11-29	2007-11-29	http://dbpedia.org/resource/Westport,_Connecticut
6	Cambridge, Mass	2007-11-29	2015-12-19	http://id.dbpedia.org/resource/Cambridge,_Massachusetts
7	London	2007-11-29	2015-12-19	http://id.dbpedia.org/resource/London
8	New York	2007-11-29	2007-11-29	http://id.dbpedia.org/resource/New_York

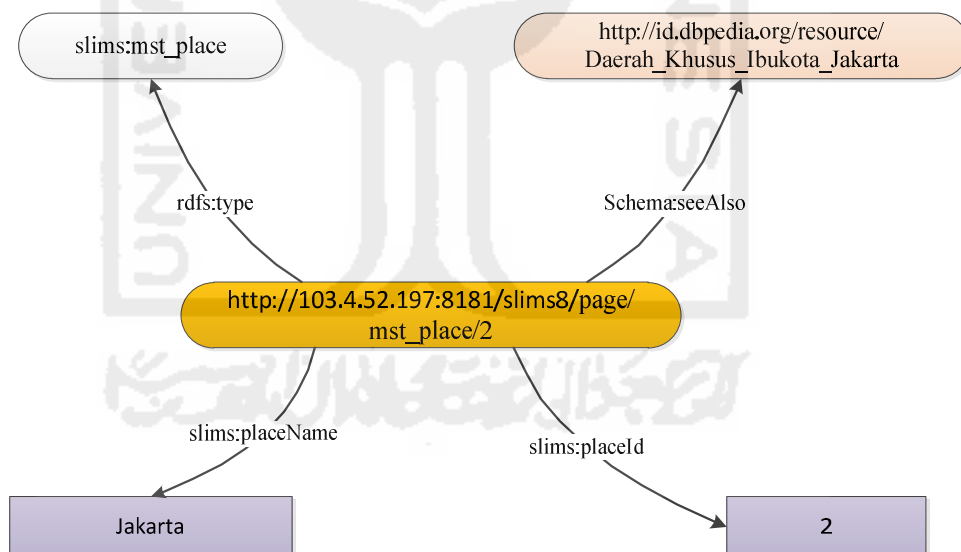
Untuk setiap entitas data *mst_place* yang tersedia akan di buat RDF yang diidentifikasi dengan URI, pada tabel 4.8 mendeskripsikan setiap predicate dan object pada RDF untuk setiap entitas class *place* yang diambil dari data *mst_place*.

Untuk properti `slims:mst_place` berisi data dari kolom `mst_place.place` dalam tabel `mst_place` dan `schema:seeAlso` yang berisi data dari kolom `mst_place.seeAlso`.

Table 4.8 RDF triple tabel `mst_place`

Predikat	Objek
<code>rdf:type</code>	<code>slims:place</code>
<code>slims:placeId</code>	<code>mst_place.place_id</code>
<code>slims:placeName</code>	<code>mst_place.place_name</code>
<code>schema:seeAlso</code>	<code>mst_place.seeAlso</code>

Pada gambar 4.7 mendeskripsikan graph RDF class `mst_place`. Properti `slims:placeName` merupakan bentuk representatif dari kolom `place_name` pada table `mst_place` dan merujuk kembali ke pada sebuah sumber data lain dengan properti `schema:seeAlso` pada class `mst_place` seperti yang terlihat pada gambar ... merujuk pada URI `http://id.dbpedia.org/resource/Daerah_Khusus_Ibukota_Jakarta` yang nantinya akan menjelaskan tentang informasi tentang data “Jakarta”, URI tersebut diambil dari *eksternal resource* `id.dbpedia.org`.



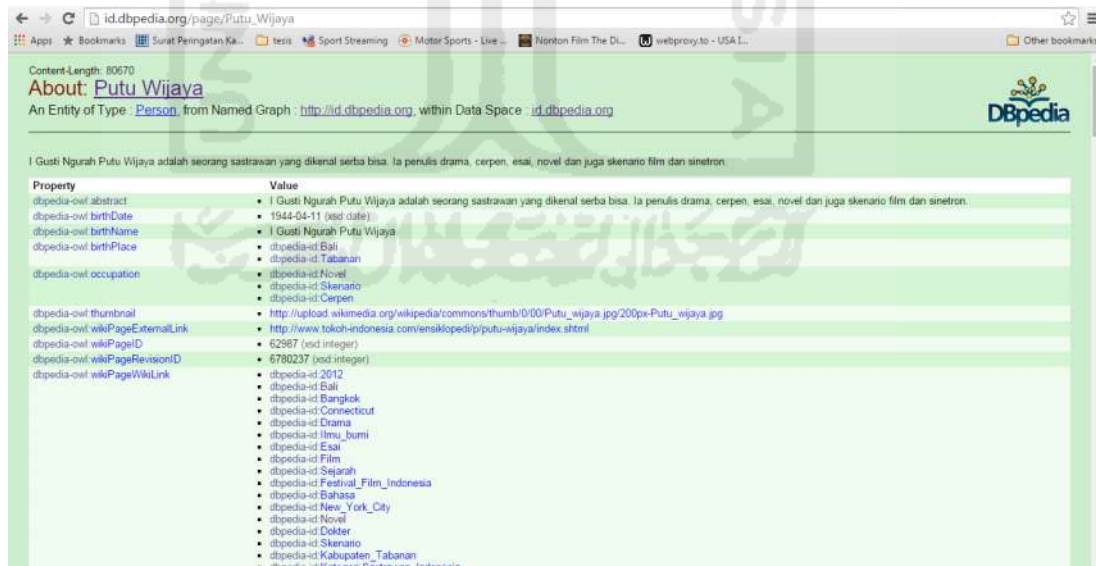
Gambar 4.7 Model graph RDF `mst_place`

4.1.6.6. Linking dengan entitas dari sumber *linked data* lain

Sebuah teknologi *linked data* tidak hanya saja mengkaitkan data-data dari internal sistem yang didefinisikan oleh URI akan tetapi juga dapat mengaitkan dengan sumber data dari sistem lain. Suatu topic atau subject karya tulis nantinya

dapat ditarik informasi yang lebih luas dengan mengkaitkan URI dari sumber data tersebut. Dengan mengkaitkan URI dari sumber data lain tersebut sistem menarik informasi melalui query pada SPARQL endpoint yang telah disediakan. Sebagai contoh graph RDF untuk data topic atau subject “W.S. Rendra” yang diidentifikasi dengan URI http://id.dbpedia.org/page/W._S._Rendra berupa informasi tentang seorang tokoh W.S. Rendra, tempat dan tanggal lahir, karya-karya sastra. Properti `rdfs:seeAlso` pada graph RDF topic merupakan bentuk representatif dari kolom `see_also` merujuk pada resource lain yang menyediakan informasi tentang sebuah topic yang ditunjukkan pada gambar 4.8 Informasi tersebut diambil dari id.dbpedia.org yang menyediakan informasi dalam bentuk *linked data*, dengan demikian informasi dapat diambil dan diintegrasikan pada sistem secara langsung melalui query SPARQL nantinya.

Tidak semua properti dbpedia akan diambil, hanya beberapa data yang diperlukan informasinya yang akan diambil, seperti properti `dbpedia-owl:abstract` yang berisi abstract dari sebuah topic, `dbpedia-owl:wikiPageWikiLink` yang berisi tautan ke link halaman Wikipedia.



The screenshot shows a web browser window with the address bar displaying id.dbpedia.org/page/Putu_Wijaya. The page content includes a title "About: Putu Wijaya" and a description: "Gusti Ngurah Putu Wijaya adalah seorang sastrawan yang dikenal serba bisa. Ia penulis drama, cerpen, esai, novel dan juga skenario film dan sinetron." Below this is a table with two columns: "Property" and "Value".

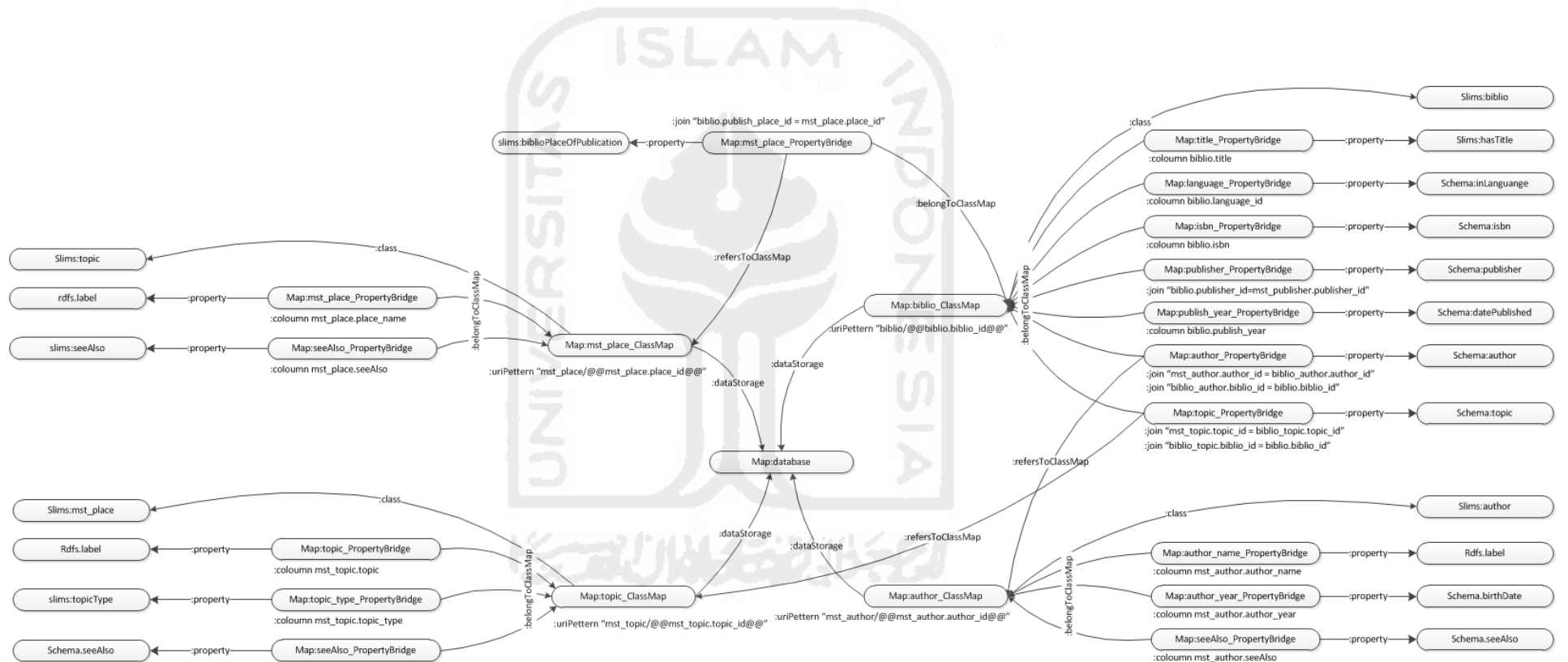
Property	Value
<code>dbpedia-owl:abstract</code>	Gusti Ngurah Putu Wijaya adalah seorang sastrawan yang dikenal serba bisa. Ia penulis drama, cerpen, esai, novel dan juga skenario film dan sinetron.
<code>dbpedia-owl:birthDate</code>	1944-04-11 (xsd:date)
<code>dbpedia-owl:birthName</code>	Gusti Ngurah Putu Wijaya
<code>dbpedia-owl:birthPlace</code>	<ul style="list-style-type: none"> dbpedia-id:Bali dbpedia-id:Tabanan
<code>dbpedia-owl:occupation</code>	<ul style="list-style-type: none"> dbpedia-id:Novel dbpedia-id:Skenario dbpedia-id:Cerpen
<code>dbpedia-owl:thumbnail</code>	http://upload.wikimedia.org/wikipedia/commons/thumb/0/00/Putu_wijaya.jpg/200px-Putu_wijaya.jpg
<code>dbpedia-owl:wikiPageExternalLink</code>	http://www.tokoh-indonesia.com/ensiklopedi/p/putu-wijaya/index.shtml
<code>dbpedia-owl:wikiPageID</code>	62987 (xsd:integer)
<code>dbpedia-owl:wikiPageRevisionID</code>	6780237 (xsd:integer)
<code>dbpedia-owl:wikiPageWikiLink</code>	<ul style="list-style-type: none"> dbpedia-id:2012 dbpedia-id:Bali dbpedia-id:Bangkok dbpedia-id:Connecticut dbpedia-id:Drama dbpedia-id:ilmu_bumi dbpedia-id:Esai dbpedia-id:Film dbpedia-id:Sejarah dbpedia-id:Festival_Film_Indonesia dbpedia-id:Bahasa dbpedia-id>New_York_City dbpedia-id:Novel dbpedia-id:Dokter dbpedia-id:Skenario dbpedia-id:Kabupaten_Tabanan dbpedia-id:Kategori_Sastrawan_Indonesia

Gambar 4.8 Data dbpedia tentang seorang tokoh Putu Wijaya

4.1.6.7. Mapping data

Untuk konversi tabel-table dalam database slims menjadi bentuk RDF dibutuhkan mapping tabel yang diimplementasikan untuk dijadikan sebuah resource, pemetaan mendefinisikan RDF graph yang berisi informasi dari database. Terlihat pada gambar 4.9 yang merupakan model diagram D2RQ mapping untuk mengimplementasikan setiap class yang telah didesain sebelumnya.





Gambar 4.9 Model diagram D2RQ mapping

Mapping mengimplementasikan setiap class yang telah didesain sebelumnya. Terdapat empat tabel dalam database “slims” yang akan digunakan, setiap tabel yang membentuk katalog bibliografi didefinisikan sebagai class map yaitu :

1. map:biblio
2. map:author
3. map:topic
4. map:mst_place

Tabel pembentuk katalog bibliografi memiliki relasi dengan tabel yang lainnya, relasi dalam mapping digambarkan `d2rq:refersToClassMap`. Class `slims:biblio` memiliki relasi atau hubungan dengan class `slims:author`, class `slims:topic` dan `slims:mst_place`, relasi hubungan ini dinyatakan dalam `d2rq:join "mst_author.author_id = biblio_author.author_id"; d2rq:join "biblio_author.biblio_id = biblio.biblio_id"`, dari pernyataan tersebut `author_id` dalam tabel `biblio_author` sama dengan `author_id` dalam tabel `mst_author` dan `biblio_id` dalam tabel `biblio_author` sama dengan `biblio_id` dalam tabel `biblio`, hal ini menggambarkan bahwa sebuah `biblio` mempunyai `author` sesuai dengan `author_id` yang terdapat dalam class `slims:author`, bentuk RDF nantinya berupa URI dari class `author` adalah `id_author`. Begitu juga dengan class `slims:biblio` dengan class `slims:topic`. Dapat diartikan bahwa sebuah `biblio` mempunyai `topic` tertentu.

Terdapat beberapa kolom tabel yang menjadi URI dalam mapping sebagai contoh terdapat pada class `slims:biblio` memiliki format URI `:uriPattern"biblio/@@biblio.biblio_id@@"`, pada class `slims:biblio` memiliki URI dari class lainnya atau dikaitkan dengan URI setiap class, untuk nantinya dapat mengambil data dari class tersebut.

Setiap class memiliki properti yang mewakili setiap kolom dalam tabel atau class. Pada class `slims:biblio` memiliki `slims:author` mendeskripsikan resource `author` yang mengacu pada class `slims:author` begitu juga untuk class yang lainnya. Pada class `slims:author`, `slims:topic` dan `slims:mst_place` diberi properti

schema:seeAlso yang nantinya berisi URI dari eksternal resource lain. Untuk setiap class dendesripsikan tabel-tabel yang diambil dari sebuah database:datastorage.

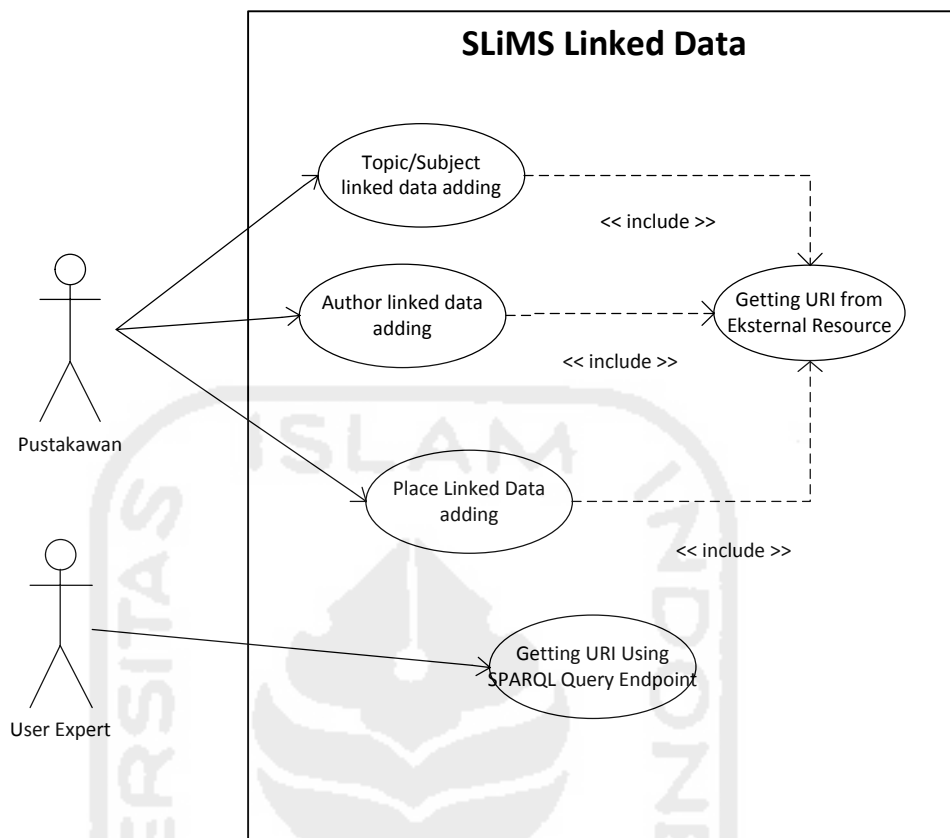
4.1.7. Rancangan Prototipe Pengembangan Aplikasi Bibliografi

4.1.7.1. Perancangan prototipe sistem

Prototipe sistem dirancang sebagai model awal sistem yang digunakan sebagai titik awal pengembangan sistem menuju sebuah sistem yang ideal bagi pengguna sistem tersebut. Perancangan prototipe sistem terbagi menjadi beberapa bagian yaitu use case digaram yang menjelaskan gambaran umum sistem, diagram aliran data untuk setiap proses yang dilakukan sistem, dan perancangan antarmuka sebagai media komunikasi sistem end-user sistem.

4.1.7.2. Use case diagram

Sistem *linked data* bibliografi mempunyai beberapa fungsi seperti yang terlihat pada Gambar 3.10, pada gambar use case tersebut terdapat aktor yaitu Pustakawan. User Expert adalah pengguna aplikasi yang memahami konsep dasar dari *linked data* atau konsep dari RDF. Dalam diagram use case aktor Admin dapat melakukan tindakan manajemen data pustaka, konfigurasi data mapping *linked data*, menambahkan link data berupa URI ke sumber eksternal lain seperti dbpedia.org, worldcat.org, viaf.org. Aktor juga bisa melihat data biblio, author, topic dalam bentuk *linked data* dengan memanfaatkan *linked data* browse atau memanfaatkan SPARQL query secara langsung menggunakan endpoint yang telah disediakan.



Gambar 4.10 Use Case SLiMS *Linked Data*

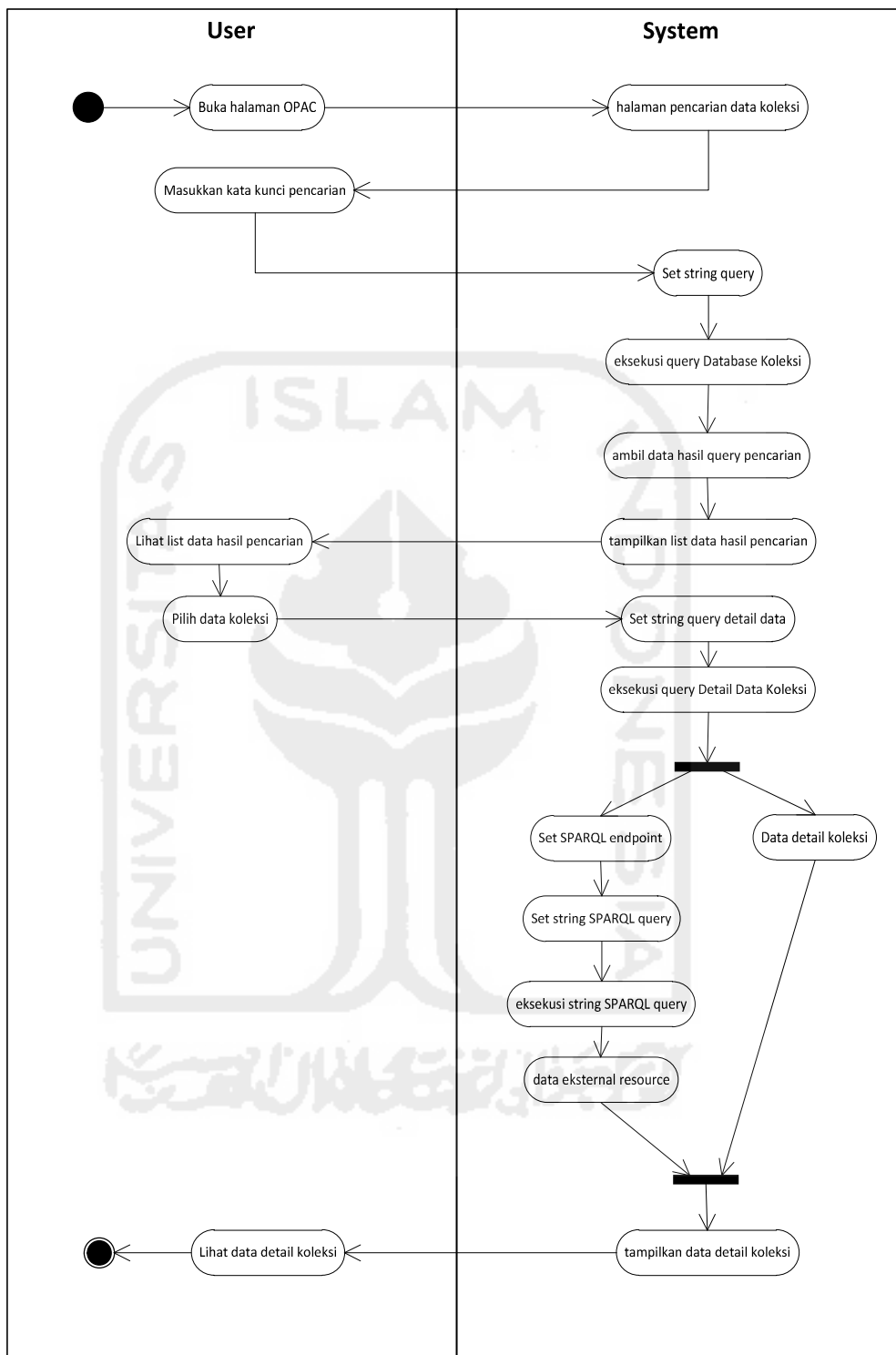
4.1.7.3. Diagram Aktifitas

Secara umum, sebuah proses yang dilakukan sistem adalah serangkaian kegiatan yang dilakukan sistem dalam rangka mengolah data pada *linked data* menggunakan aturan-aturan tertentu untuk menghasilkan sebuah informasi. Banyaknya proses yang dapat dilakukan sebuah sistem dipengaruhi oleh banyaknya feature atau fasilitas yang disediakan sistem tersebut. Mengacu pada use case diagram yang telah terbentuk. Terdapat dua proses utama yaitu proses menampilkan data bibliografi dan menambahkan URI *linked data* bibliografi. SPARQL endpoint dimana user dapat melakukan query langsung terhadap data yang disediakan, kemudian user dapat mendownload data dalam bentuk RDF N-TRIPLE.

Untuk proses yang pertama adalah menampilkan data bibliografi dalam bentuk *linked data* seperti yang ditunjukkan pada gambar diagram 4.11 Di dalam aplikasi SLiMS sudah terdapat proses menampilkan data bibliografi dan proses

pencarian data bibliografi. Dimana proses tersebut menggunakan SQL query guna mendapatkan data dari database yang telah disiapkan. Dalam rancangan pengembangan aplikasi SLiMS, proses menampilkan data bibliografi dikembangkan dengan menampilkan data bibliografi dalam bentuk *linked data*. Informasi yang di tampilkan adalah informasi dalam bentuk *linked data* dengan menampilkan informasi tambahan dalam bentuk URI.

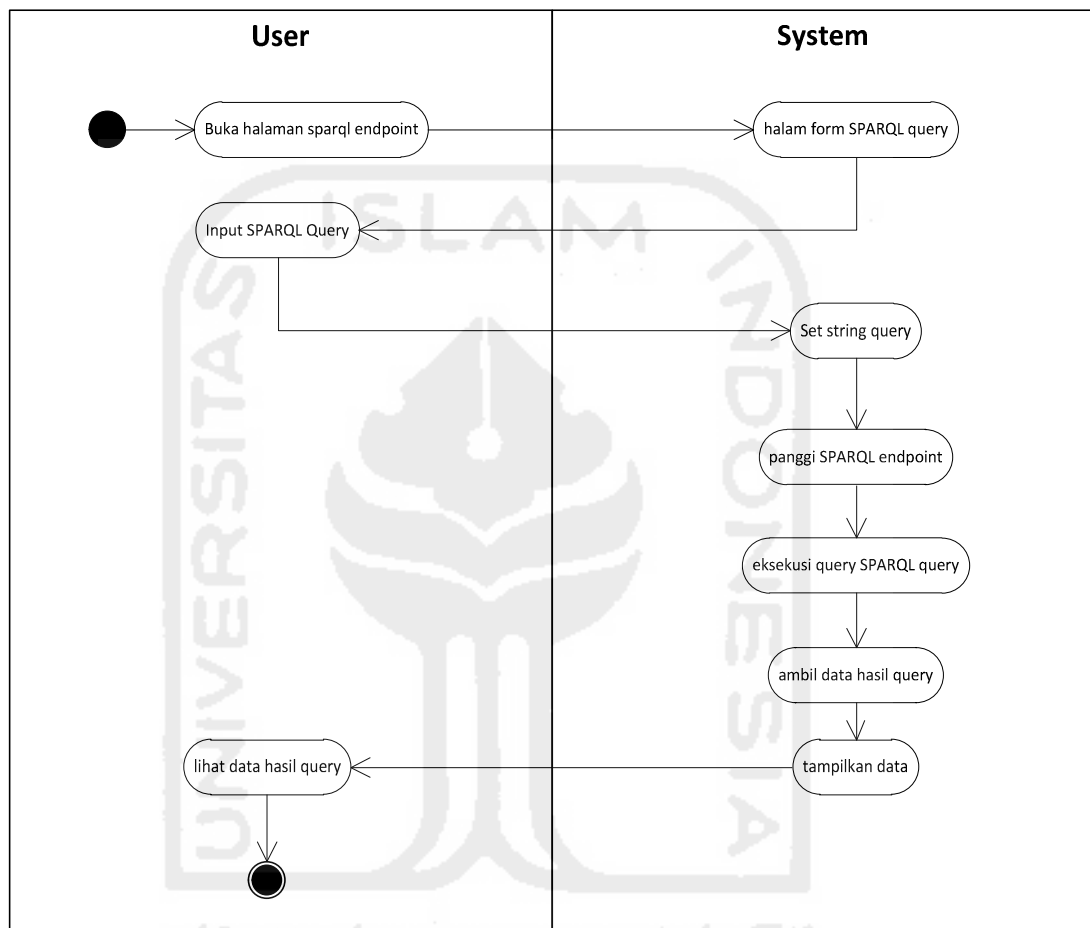




Gambar 4.11 Diagram aktifitas menampilkan data bibliografi

Setiap user juga dapat melihat data-data yang disediakan dengan cara melakukan query langsung yang disediakan dalam halam SPARQL *Endpoint*. Dalam

gambar 4.12 diagram aktivitas Query SPARQL Endpoint ketika form query terbuka, user dapat mengetikkan perintah-perintah query SPARQL untuk menquery data yang tersedia. String query dieksekusi oleh SPARQL Endpoint yang disediakan sistem, kemudia hasil dari query akan ditampilkan.

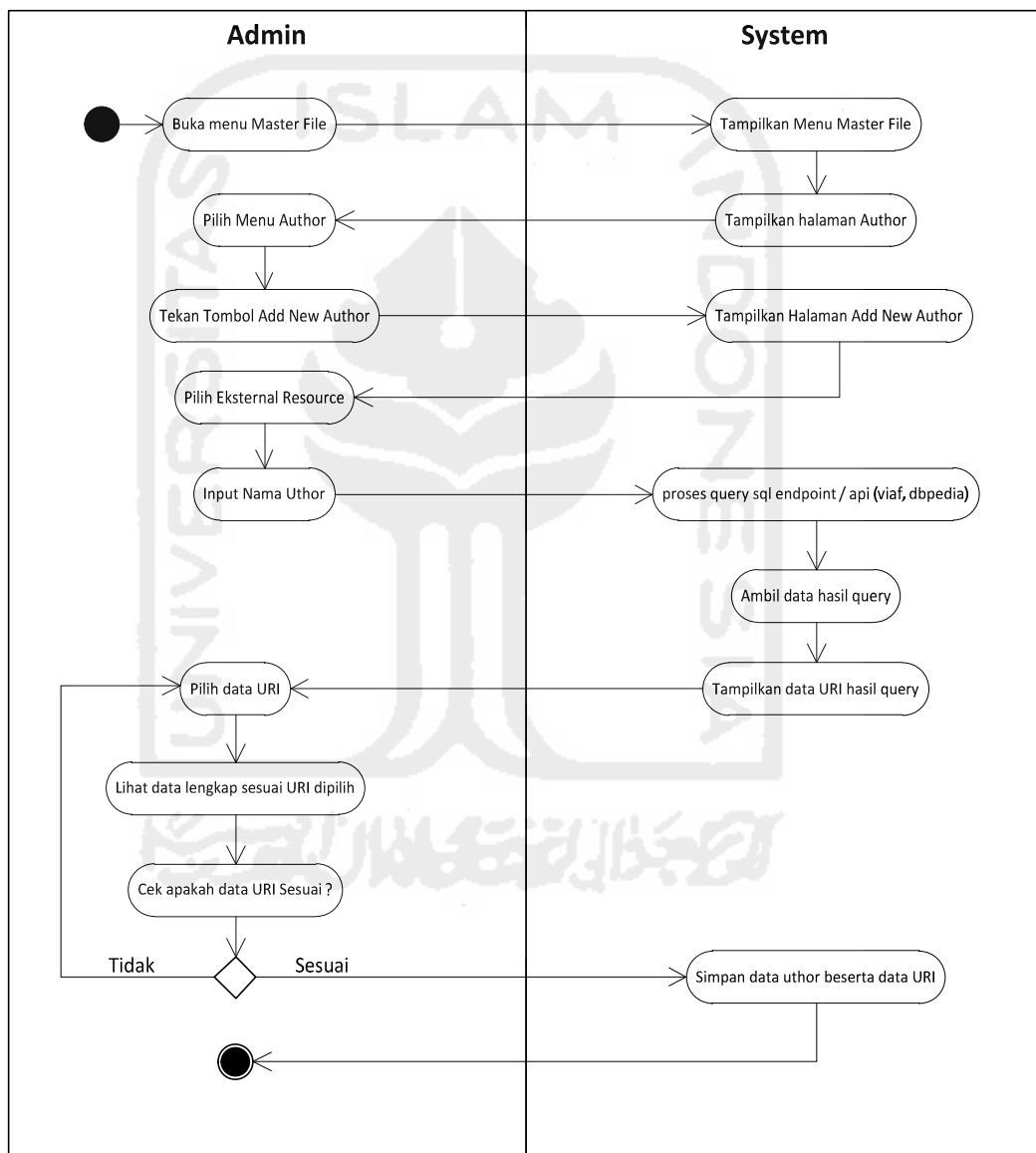


Gambar 4.12 Diagram aktifitas query Sparql

Pada aplikasi SLiMS sebelumnya telah tersedia fungsi manajemen data bibliografi dan transaksi data pada perpustakaan. Semua fungsi tersebut dapat digunakan oleh user yang memiliki hak akses khusus dimana user diwajibkan melakukan proses login terlebih dahulu untuk bisa menggunakan fungsi-fungsi tersebut. Rancangan pengembangan pada aplikasi SLiMS adalah menambahkan fungsi *linked data* pada data bibliografi.

Pada gambar 4.13 Menggambarkan diagram aktivitas untuk melakukan tambah data URI untuk mengaitkan data biblio, author, topic ke resource eksternal

lainnya. Proses penambahan URI disini admin akan memilih data apa yang akan ditambahkan URI. Ketika admin memilih topic yang akan di tambahkan data URI maka sistem akan melakukan query SPARQL Endpoint ke resource *linked data* eksternal untuk menampilkan informasi mengenai data topic yang di cari. Admin dapat memilih data topic yang menurut admin relefan dengan data topic yang dicari dan disimpan oleh sistem.



Gambar 4.13 Diagram aktifitas menambahkan URI *linked data*

Data-data bibliografi yang awalnya disimpan dalam database dapat dikonversi kedalam bentuk RDF triple. Untuk melakukan konversi data dibutuhkan

mapping file yang nantinya akan dieksekusi oleh D2RQ. Pembuatan mapping perlu memperhatikan tabel-tabel yang terkait dengan data bibliografi dalam database SLiMS seperti yang telah dijelaskan pada bab sebelumnya. Untuk melakukan perubahan atau mengkonfigurasi file mapping disediakan fungsi halaman edit mapping configuration oleh sistem.

4.1.7.4. Perancangan Antarmuka

Pembuatan antarmuka sistem merupakan sebuah media komunikasi sistem dengan end-user sistem. Secara umum model yang digunakan dalam perancangan antarmuka pada sistem *Linked Data* Bibliografi mengacu pada model aplikasi SLiMS yang telah menggunakan website sebagai antarmuka. Sebuah website pada umumnya terdiri dari homepage yang dilengkapi dengan berbagai menu yang berisikan link sesuai dengan fungsi masing-masing-masing. Berikut adalah visualisasi pengembangan aplikasi SLiMS menggunakan teknologi *linked data*. Pada perancangan antarmuka ini, tidak semua proses pada aplikasi SLiMS dilakukan pengembangan. Pengembangan aplikasi SLiMS menggunakan teknologi *linked data* meliputi beberapa bagian yaitu :

1. Bagian proses master file Author
2. Bagian proses master file Subject/Topic
3. Bagian proses master file Place
4. Bagian proses bibliografi
5. Bagian proses penelusuran data bibliografi

Pada bagian master file Author dirancang pengembangan pada proses input data dan edit data Author yang ditunjukkan pada gambar 4.14, 4.15 dan gambar 4.16. Pada bagian ini user admin dapat melakukan proses penambahan data link dari eksternal resource lain. Pada rancangan ini user admin dapat mengetikkan nama seorang author dilanjutkan dengan menekan tombol “Get URI” untuk melakukan proses pendarian data. Dari hasil pencarian sistem akan menampilkan beberapa pilihan nama author yang bisa dipilih. Setelah user admin memilih salah satu author sistem akan menampilkan data URI dan informasi tambahan tentang Author yang dipilih.

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Author

Search :

Paging : 1 2 3 4 5 6 Next Last Page

Del	Edit	Author Name	Year	See Also
<input type="checkbox"/>	Text	Text	Text	Text
<input type="checkbox"/>	Text	Text	Text	Text
<input type="checkbox"/>	Text	Text	Text	Text

Gambar 4.14 Rancangan interface list data author

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Author

Eksternal Source :

Author Name :

Search Result (URI) :

Author Birth Year :

Authority Type :

Authority files :

URI – Eksternal Resource :

Gambar 4.15 Rancangan interface tambah data URI *linked data author*

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Linked Data Author

Search :

Paging : 1 2 3 4 5 6 Next Last Page

Linking	Author Name	Year	See Also
<input type="checkbox"/> Text	Text	Text	Text
<input type="button" value="Remove Link"/>	Text	Text	Text
<input type="checkbox"/> Text	Text	Text	Text

Gambar 4.16 Rancangan interface tambah data

URI *linked data author* secara kolektif

Perancangan halaman proses selanjutnya adalah halaman proses master file Subject/Topic yang ditunjukkan pada gambar 4.17 dan 4.18 Pada rancangan bagian ini user admin dapat melakukan proses penambahan dan perubahan data link subject/topic dari eksternal resource lain. Disediakan form pilihan dimana user admin dapat memilih eksternal resource *linked data* yang di inginkan. Kemudian dilanjutkan dengan mengetikan data subject/topic dilanjutkan dengan menekan tombol “Get URI” untuk melakukan proses pencarian data dari *eksternal resource* (id.dbpedia.org atau dbpedia.org). Hasil pencarian akan ditampilkan dalam bentuk pilihan. Setelah user admin memilih salah satu hasil pencarian data subject/topic sistem akan menampilkan data URI beserta informasi tambahan tentang subject yang dimaksud.

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Subject

Search :

Paging : 1 2 3 4 5 6 Next Last Page

Del	Edit	Subject Name	Year	See Also
<input type="checkbox"/>	Text	Text	Text	Text
<input type="checkbox"/>	Text	Text	Text	Text
<input type="checkbox"/>	Text	Text	Text	Text

Gambar 4.17 Rancangan interface list data subject atau topic

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Subject

Eksternal Source :

Subject :

Search Result:

Classification Code :

Subject Type :

Authority files :

URI – Eksternal Resource :

URI Information :

Gambar 4.18 Rancangan interface tambah data URI *linked data* subject

Perancangan halaman proses selanjutnya adalah halaman proses master file *place* yang ditunjukkan pada gambar 4.19 dan 4.20 Pada rancangan bagian ini user

admin dapat melakukan proses penambahan dan perubahan data URI place dari eksternal resource lain. Disediakan form pilihan dimana user admin dapat memilih eksternal resource *linked data* yang di inginkan yaitu dbpedia.org dan id.dbpedia.org. Kemudian dilanjutkan dengan mengetikan data *place* dilanjutkan dengan menekan tombol “Get URI” untuk melakukan proses pencarian data dari *eksternal resource* (id.dbpedia.org atau dbpedia.org). Hasil pencarian akan ditampilkan dalam bentuk pilihan. Setelah user admin memilih salah satu hasil pencarian data *place* sistem akan menampilkan data URI beserta informasi tambahan tentang subject yang dimaksud.

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Place

Search :

Paging : 1 2 3 4 5 6 Next Last Page

Del	Edit	Place Name	Year	See Also
<input type="checkbox"/>	Text	Text	Text	Text
<input type="checkbox"/>	Text	Text	Text	Text
<input type="checkbox"/>	Text	Text	Text	Text

Gambar 4.19 Rancangan interface list data *place*

[Main Menu 1] [Main Menu 2] [Main Menu 3]

[Sub Menu 1]
[Sub Menu 2]
[Sub Menu 3]

Place

Eksternal Source :

Place Name :

Search Result:


URI – Eksternal Resource :

URI Information :

Gambar 4.20 Rancangan interface tambah data URI *linked data* place

Kemudian selanjutnya adalah perancangan pada halaman penelusuran data bibliografi. Pada 4.21, aplikasi SLiMS sebelumnya telah tersedia fasilitas proses penelusuran data bibliografi. Pada proses penelusuran data bibliografi disediakan dua macam jenis pencarian yaitu pencarian sederhana dan pencarian lengkap. Perbedaan dari kedua jenis pencarian tersebut adalah pada pencarian lengkap disediakan beberapa kriteria pencarian yang lebih spesifik seperti pencarian data berdasarkan author, subject, ISBN, dan jenis koleksi. Untuk rancangan pengembangan pada proses penelusuran data bibliografi yang dikaitkan dengan teknologi *linked data* adalah pada halaman detail bibliografi dimana informasi yang di sajikan pada halaman detail bibliografi selain menampilkan informasi bibliografi yang dimaksud juga akan menyajikan data tambahan dari eksternal resource lain.

[Main Menu 1] [Main Menu 2] [Main Menu 3]



Judul Koleksi Data

Author(s) : Authors 1
(Linked Data : <http://viaf.org/viaf/xxxxxx>)
Work :
- Judul artikel 1
- Judul artikel 2

Editon :
Call Number :
ISBN/ISSN :
Subject : Subject 1 ([linked data dbpedia.org/wiki/xxxxxx](http://dbpedia.org/wiki/xxxxxx))
Subject 2 ([linked data dbpedia.org/wiki/xxxxxx](http://dbpedia.org/wiki/xxxxxx))

Classification :
Series Title :
GMD :
Language :
Publisher :
Publishing Year :
Publishing Place :
Collation :
File Attachment :
Avability :

Linked Data (<http://...../resource/biblio/xxxx>)

Property	Value
xxxxxxx	xxxxxxxxxxx
xxxxxxx	xxxxxxxxxxx
xxxxxxx	xxxxxxxxxxx

Information

Wikipedia

Wikipedia link

Gambar 4.21 Rancangan interface detail data bibliografi

4.2. Implementasi

Pada penelitian ini, pembuatan repository *linked data* bibliografi diimplementasikan dengan menggunakan bahasa pemrograman PHP. Implementasi pengembangan sistem ini sesuai dengan arsitektur yang telah dirancang pada bab sebelumnya.

4.2.1. Implementasi pembuatan mapping

Pembuatan mapping class, properties dan instance dari tabel-tabel database dalam RDF menggunakan Notepad dan disimpan ke dalam file `slimsbiblio.n3`. File ini yang nantinya akan dijalankan oleh D2RQ untuk menampilkan data dari database ke bentuk RDF N-TRIPLE, Gambar 4.22 menunjukkan namespace dari proses mapping

```
@prefix map: <#> .
@prefix db: <> .
```

```

@prefix slims: <slims/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
@prefix schema: <http://schema.org/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix      d2r:      <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-
server/config.rdf#> .

```

Gambar 4.22 Namespace mapping

4.2.1.1. Pembuatan koneksi ke database

Dalam pembuatan mapping diperlukan sumber data dari database, sedangkan untuk bisa mengakses data tersebut maka dibutuhkan suatu koneksi ke dalam database “slims”. Gambar 4.23 menunjukkan mapping koneksi ke dalam database.

```

map:database a d2rq:Database;
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:jdbcDSN "jdbc:mysql://localhost/senayan-isi ";
    d2rq:username "root";
    jdbc:autoReconnect "true";
    jdbc:zeroDateTimeBehavior "convertToNull";

```

Gambar 4.23 Mapping koneksi database

4.2.1.2. Pembuatan property dan instance pada class biblio

Dalam pembentukan mapping setiap tabel dalam database bibliografi dijadikan class, pada gambar 4.24 pada tabel biblio didefinisikan sebagai class biblio. Pada properti slims:biblio_publisher_id terlihat bahwa instance diambil dari proses join dimana biblio.publisher_id=mst_publisher.publisher_id dalam hubungan antar tabel bisa diartikan bahwa banyak data bibliografi mempunyai publisher(penerbit) tertentu. Begitu juga dengan properti schema:author dimana instance diambil dari dua proses join yaitu relasi antara tabel mst author dengan table biblio_athor yang memiliki relasi mst_author.author_id = biblio_author.author_id dan relasi berikutnya

antara tabel `biblio_author` dengan tabel `biblio` yang memiliki relasi `biblio_author.author_id = biblio.biblio_id`.

```
# Table biblio
map:biblio a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "biblio/@@biblio.biblio_id@@";
  d2rq:class slims:biblio;
  d2rq:classDefinitionLabel "biblio";
.
map:biblio_label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property rdfs:label;
  d2rq:pattern "^^biblio.title^^";
.
map:biblio_biblio_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_biblio_id;
  d2rq:propertyDefinitionLabel "biblio biblio_id";
  d2rq:column "biblio.biblio_id";
  d2rq:datatype xsd:integer;
.
map:biblio_gmd_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_gmd_id;
  d2rq:propertyDefinitionLabel "biblio gmd_id";
  d2rq:column "biblio.gmd_id";
  d2rq:datatype xsd:integer;
.
map:biblio_title a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property schema:name;
  d2rq:propertyDefinitionLabel "biblio title";
  d2rq:column "biblio.title";
.
map:biblio_sor a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_sor;
  d2rq:propertyDefinitionLabel "biblio sor";
  d2rq:column "biblio.sor";
.
map:biblio_edition a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_edition;
  d2rq:propertyDefinitionLabel "biblio edition";
  d2rq:column "biblio.edition";
.
map:biblio_isbn_issn a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property schema:isbn;
  d2rq:propertyDefinitionLabel "biblio isbn_issn";
  d2rq:column "biblio.isbn_issn";
.
```

```

map:biblio_publish_year a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property schema:datePublished;
  d2rq:propertyDefinitionLabel "biblio publish_year";
  d2rq:column "biblio.publish_year";
.
map:biblio_collation a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_collation;
  d2rq:propertyDefinitionLabel "biblio collation";
  d2rq:column "biblio.collation";
.
map:biblio_series_title a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_series_title;
  d2rq:propertyDefinitionLabel "biblio series_title";
  d2rq:column "biblio.series_title";
.
map:biblio_call_number a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_call_number;
  d2rq:propertyDefinitionLabel "biblio call_number";
  d2rq:column "biblio.call_number";
.
map:biblio_language_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property schema:inLanguage;
  d2rq:propertyDefinitionLabel "biblio language_id";
  d2rq:column "biblio.language_id";
.
map:biblio_source a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_source;
  d2rq:propertyDefinitionLabel "biblio source";
  d2rq:column "biblio.source";
.
map:biblio_publish_place_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_publish_place_id;
  d2rq:propertyDefinitionLabel "biblio publish_place_id";
  d2rq:column "biblio.publish_place_id";
  d2rq:datatype xsd:integer;
.
map:biblio_classification a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_classification;
  d2rq:propertyDefinitionLabel "biblio classification";
  d2rq:column "biblio.classification";
.
map:biblio_notes a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:biblio;
  d2rq:property slims:biblio_notes;
  d2rq:propertyDefinitionLabel "biblio notes";
  d2rq:column "biblio.notes";
.
map:biblio_image a d2rq:PropertyBridge;

```

```

d2rq:belongsToClassMap map:biblio;
d2rq:property slims:biblio_image;
d2rq:propertyDefinitionLabel "biblio image";
d2rq:column "biblio.image";
.
map:biblio_file_att a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:biblio;
d2rq:property slims:biblio_file_att;
d2rq:propertyDefinitionLabel "biblio file_att";
d2rq:column "biblio.file_att";
.
map:biblio_promoted a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:biblio;
d2rq:property slims:biblio_promoted;
d2rq:propertyDefinitionLabel "biblio promoted";
d2rq:column "biblio.promoted";
d2rq:datatype xsd:integer;
.
map:biblio_labels a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:biblio;
d2rq:property slims:biblio_labels;
d2rq:propertyDefinitionLabel "biblio labels";
d2rq:column "biblio.labels";
.
map:biblio_publisher_id_ref a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:biblio;
d2rq:property slims:biblio_publisher_id;
d2rq:refersToClassMap map:mst_publisher;
d2rq:join"biblio.publisher_id =>
mst_publisher.publisher_id";
.
# n:m table rel author biblio
map:rel_author_biblio a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:biblio;
d2rq:property schema:author;
d2rq:refersToClassMap map:mst_author;
d2rq:join "mst_author.author_id = biblio_author.author_id";
d2rq:join "biblio.author.author_id = biblio.biblio_id";

```

Gambar 4.24 Diagram mapping table biblio ke class biblio

4.2.1.3. Pembuatan property dan instance pada class mst_author

Pada tahap berikutnya adalah pembuatan class, property dan instance pada tabel mst_author seperti yang ditunjukkan pada gambar 4.25 Sebagai URI (Uniform Resource Identifier) adalah author_id dengan memberikan kode baris pada pada file mapping yaitu :

d2rq:uriPattern "mst_author / @@mst_author.author_id";

Selanjutnya untuk properti label pada class mst_author, rdfs:label data diambil dari kolom author_id. Untuk pembentukan properti

slims:mst_author_author_name instance data berasal dari kolom mst_author.author_name.

```
# Table mst_author
map:mst_author a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "mst_author/@@mst_author.author_id@";
    d2rq:class slims:mst_author;
    d2rq:classDefinitionLabel "mst_author";.
map:mst_author__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:mst_author;
    d2rq:property rdfs:label;
    d2rq:pattern "mst_author #@@mst_author.author_id@";.
map:mst_author_author_id a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:mst_author;
    d2rq:property slims:mst_author_author_id;
    d2rq:propertyDefinitionLabel "mst_author author_id";
    d2rq:column "mst_author.author_id";
    d2rq:datatype xsd:integer;.
map:mst_author_author_name a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:mst_author;
    d2rq:property slims:mst_author_author_name;
    d2rq:propertyDefinitionLabel "mst_author author_name";
    d2rq:column "mst_author.author_name";.
map:mst_author_author_year a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:mst_author;
    d2rq:property slims:mst_author_author_year;
    d2rq:propertyDefinitionLabel "mst_author author_year";
    d2rq:column "mst_author.author_year";.
map:mst_author_authority_type a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:mst_author;
    d2rq:property slims:mst_author_authority_type;
    d2rq:propertyDefinitionLabel "mst_author authority_type";
    d2rq:column "mst_author.authority_type";
```

Gambar 4.25 Diagram mapping table mst_author ke class mst_author

4.2.1.4. Pembuatan property dan instance pada class mst_topic

Tahap selanjutnya adalah pembuatan class, property dan instance pada tabel mst_topic seperti yang ditunjukkan pada gambar 4.26 sebagai URI (Uniform

Resource Identifier) adalah `topic_id` dengan memberikan kode baris pada pada file mapping yaitu :

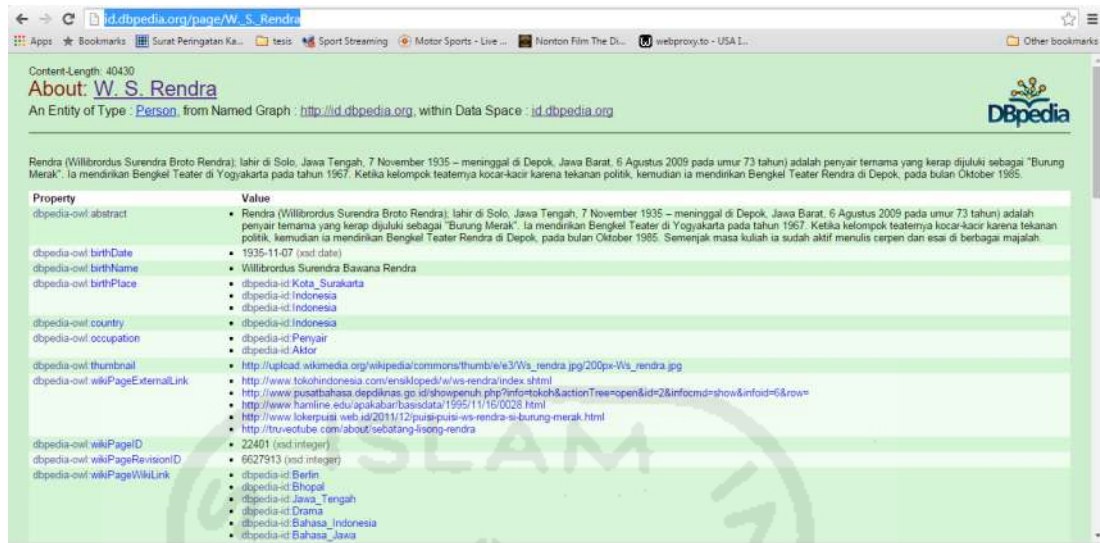
d2rq:uriPattern “mst_topic / @@mst_topic.topic_id”;

Selanjutnya untuk properti label pada class `mst_topic`, `rdfs:label` data diambil dari kolom `author_id.topic`. Untuk pembentukan properti `slims:mst_topic_seeAlso` instance data berasal dari kolom `mst_topic.see_also` properti ini adalah tautan ke eksternal resource lain yaitu salah satunya adalah `dbpedia.org`.

```
# Table mst_topic
map:mst_topic a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "mst_topic/@@mst_topic.topic_id@";
  d2rq:class slims:mst_topic;
  d2rq:classDefinitionLabel "mst_topic";.
map:mst_topic_label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:mst_topic;
  d2rq:property rdfs:label;
  d2rq:pattern "@@mst_topic.topic@";
map:mst_topic_topic_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:mst_topic;
  d2rq:property slims:mst_topic_topicId;
  d2rq:propertyDefinitionLabel "mst topic topic id";
  d2rq:column "mst_topic.topic_id";
  d2rq:datatype xsd:integer;.
map:mst_topic_topic_type a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:mst_topic;
  d2rq:property slims:mst_topic_topicType;
  d2rq:propertyDefinitionLabel "mst topic topic_type";
  d2rq:column "mst_topic.topic_type";
map:mst_topic_seeAlso a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:mst_topic;
  d2rq:property slims:mst_topic_seeAlso;
  d2rq:propertyDefinitionLabel "mst_topic seeAlso";
  d2rq:column "mst_topic.see_also";
```

Gambar 4.26 Diagram mapping table `mst_topic` ke class `mst_topic`

Sebagai contoh tautan data kolom `mst_topic.topic` ke URI dengan alamat http://id.dbpedia.org/page/W._S._Rendra untuk mendapatkan informasi tambahan yang berkaitan dengan sebuah topic “W. S. Rendra” seperti yang ditunjukkan pada gambar 4.27



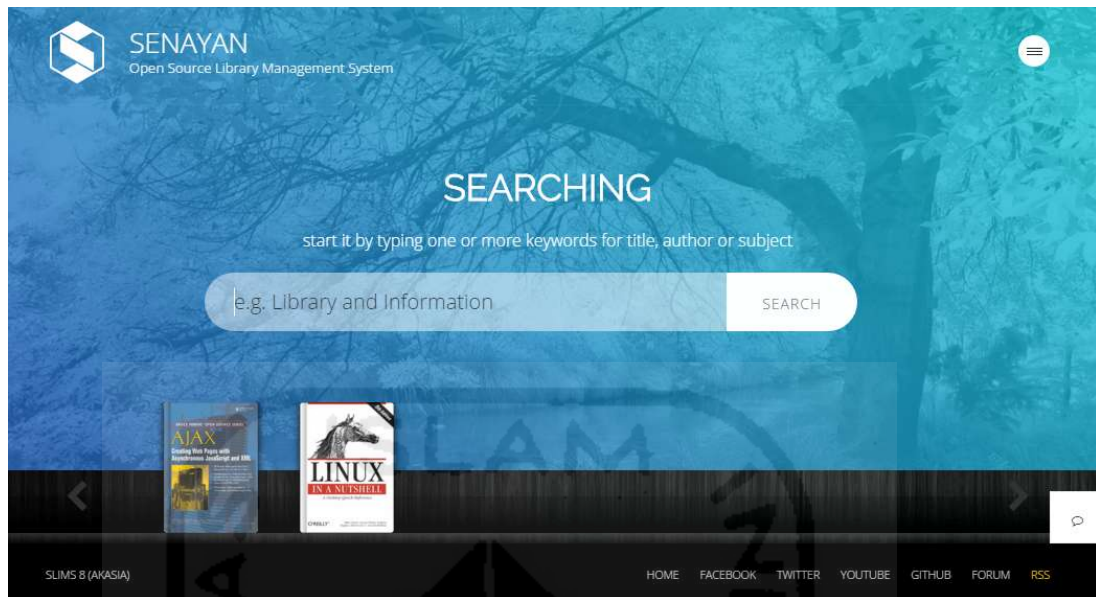
Gambar 4.27 Halaman *linked data* pada dbpedia

4.2.2. Implementasi Pembuatan Prototipe Aplikasi

Pada tahap implementasi pembuatan *prototipe* aplikasi terdapat beberapa bagian utama yaitu bagian modul menampilkan data koleksi dan bagian halaman admin. Pada bagaian modul pencarian datakoleksi, data dari hasil proses mapping bibliografi *linked data* dilakukan query sehingga data tersebut bisa ditampilkan dan informasinya dapat dibaca langsung dalam format HTML, sedangkan pada halaman admin terdapat beberapa sub-bagian modul yaitu modul untuk penambahan data link URI author dan subject.

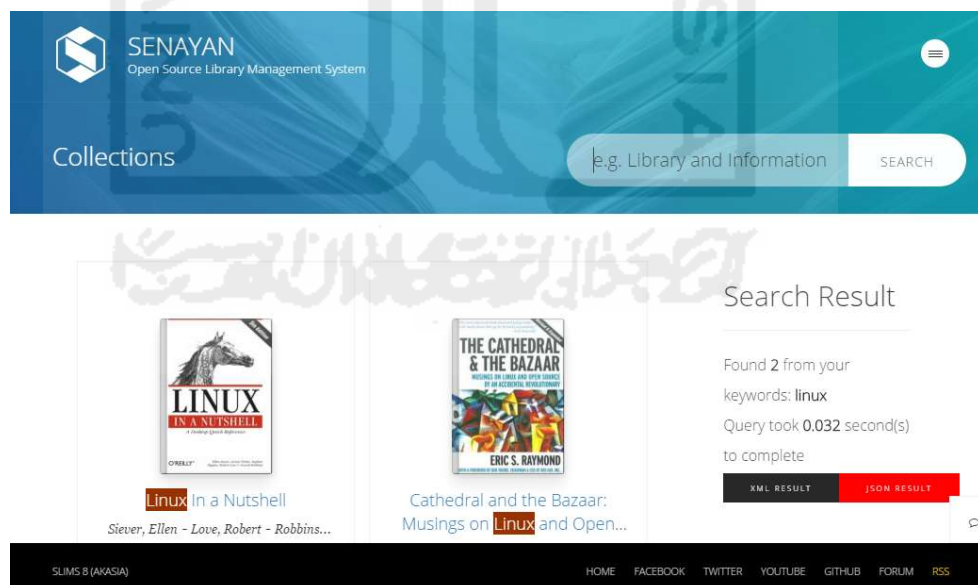
4.2.2.1. Modul Menampilkan Data Koleksi

Modul menampilkan data koleksi merupakan modul utama dari sisi pengguna atau pengunjung perpustakaan. Pada aplikasi SLiMS sebelumnya telah tersedia fungsi tersebut yang memberikan kemudahan kepada pengunjung perpustakaan untuk melakukan penelusuran data koleksi pustaka. Halaman ini sering disebut sebagai halaman OPAC (*Online Public Access Catalog*). Halaman ini dapat dilihat pada gambar 4.28 Pada halaman ini disediakan fungsi pencarian data koleksi sederhana dan advanced search.



Gambar 4.28 Halaman pencarian data pada SLiMS

Hasil proses pencarian data koleksi pustaka seperti yang terlihat pada gambar 4.29 tetap menggunakan fungsi yang telah digunakan oleh aplikasi SLiMS sebelumnya, dimana proses pencarian dengan melakukan SQL query secara langsung terhadap database MySQL yang telah disiapkan.



Gambar 4.29 Halaman list data hasil pencarian koleksi

Implementasi pengembangan aplikasi SLiMS yang penulis lakukan adalah pada proses menampilkan detail koleksi pustaka. Proses dimulai dengan user

memilih salah satu judul koleksi dari hasil pencarian data. Kemudian proses akan dilanjutkan ke halaman detail koleksi pustaka sesuai dengan data judul yang dipilih sebelumnya. Halaman ini dapat dilihat pada gambar 4.30 Pada halaman detail koleksi pustaka selain menampilkan data dari hasil query ke database serta menampilkan informasi tambahan menggunakan *linked data* bibliografi.

Text

Linux In A Nutshell

Most widely held works by authors - www.worldcat.org

- [Siever, Ellen - Personal Name](#)
- [Love, Robert - Personal Name](#)
- [Robbins, Arnold - Personal Name](#)
- [Figgins, Stephen - Personal Name](#)
- [Weber, Aaron - Personal Name](#)

Description Not Available

Information

RECORD DETAIL

BACK TO PREVIOUS

XML DETAIL CITE THIS OTHERS LIBRARIES

Subject Information

Sistem operasi adalah seperangkat program yang mengelola sumber daya perangkat keras komputer, dan menyediakan layanan umum untuk aplikasi perangkat lunak. Sistem operasi

ASIA) HOME FACEBOOK TWITTER YOUTUBE GITHUB FORUM

Gambar 4.30 Halaman detail koleksi pustaka

Informasi *linked data* yang ditampilkan meliputi data author yang di hubungkan dengan *eksternal resource* dari VIAF.org, data subject yang di hubungkan dengan *eksternal resource* dari id.dbpedia.org, dbpedia.org dan wikipedia.org seperti yang ditunjukkan pada gambar 4.31

Belum masuk log Pembicaraan IP ini Contributions Buat akun baru Masuk log

Halaman Pembicaraan Baca Sunting Sunting sumber Versi terdahulu Cari

Sistem operasi

[tutup]

Dari Wikipedia bahasa Indonesia, ensiklopedia bebas

Artikel bertopik teknologi informasi ini perlu dirapikan agar memenuhi standar Wikipedia

Merapikan artikel bisa berupa membagi artikel ke dalam paragraf atau wikifikasi artikel. Setelah dirapikan, tolong hapus pesan ini.

Sistem operasi (bahasa Inggris: ***operating system***; **OS**) adalah komponen pengolah piranti lunak dasar (*essential component*) tersistem sebagai pengelola sumber daya perangkat keras komputer (hardware), dan menyediakan layanan umum untuk aplikasi perangkat lunak. Sistem operasi adalah jenis yang paling penting dari perangkat lunak sistem dalam sistem komputer. Tanpa sistem operasi, pengguna tidak dapat menjalankan program aplikasi pada komputer mereka, kecuali program booting.

Sistem operasi mempunyai penjadwalan yang sistematis mencakup perhitungan penggunaan memori, pemrosesan data, penyimpanan data, dan sumber daya lainnya. Untuk fungsi-fungsi perangkat keras seperti sebagai masukan dan keluaran dan alokasi memori, sistem operasi bertindak sebagai perantara antara program aplikasi dan perangkat keras komputer.^{[1][2]} meskipun kode aplikasi biasanya dieksekusi langsung oleh perangkat keras dan seringkali akan menghubungi OS atau terputus oleh itu. Sistem operasi yang

Sistem operasi

```

graph TD
    Pengguna --> Aplikasi
    Aplikasi --> Sistem_operasi[Sistem operasi]
    Sistem_operasi --> Perangkat_keras[Perangkat keras]
    Perangkat_keras --> Sistem_operasi
    Sistem_operasi --> Aplikasi
    Aplikasi --> Pengguna
  
```

Gambar 4.31 Halaman tautan dari wikipedia.org

Selain itu ditambahkan juga data judul koleksi pustaka lainnya hasil dari *linked data* author yang bersumber dari *eksternal resource* VIAF.org serta sekaligus memberikan tautan link ke halaman web site Worldcat.org. Hal ini dilakukan bertujuan untuk bisa memberikan informasi data bibliografi secara lengkap sesuai judul yang dipilih seperti yang dapat dilihat pada gambar 4.32

Home Search Create lists, bibliographies and reviews: Sign in or create a free account

WorldCat

Advanced Search Find a Library

<< Return to Search Results Cite/Export Print E-mail Share Permalink

Add to list Add tags Write a review Rate this item: ☆☆☆☆

Linux kernel development

Author: [Robert Love](#)

Publisher: Indianapolis, Ind. : Novell Press, ©2005.

Edition/Format: eBook : Document : English : 2nd ed [View all editions and formats](#)

Database: WorldCat

Summary: "Linux Kernel Development details the design and implementation of the Linux kernel, presenting the content in a manner that is beneficial to those writing and developing kernel code, as well as to programmers seeking to better understand the operating system and become more efficient and productive in their coding. The book details the major subsystems and features of the Linux kernel, including its design. [Read more...](#)

Rating: ☆☆☆☆ (not yet rated) 0 with reviews - Be the first.

Subjects: [Linux](#), [Operating systems \(Computers\)](#), [Linux \(Logiciels\)](#), [View all subjects](#)

More like this [Similar items](#)

Nearby libraries

- STMIK AMIKOM
Yogyakarta
Sleman 555283,
Indonesia
3m / 3.6km
- STIKES Bethesda
Yakkum
Yogyakarta 55224,
Indonesia
5m / 8.0km
- Universitas Sanata
Dharma

Librarian? [Claim your library](#)

Gambar 4.32 Halaman tautan pada worldcat.org

Potongan kode untuk fungsi menampilkan data koleksi pustaka terlihat pada gambar 4.33 Yang terdapat fungsi query sparql untuk menampilkan data dari

eksternal resource seperti dbpedia.org dan id.dbpedia.org. Pada potongan kode tersebut terdapat fungsi query sparql untuk menampilkan data deskripsi atau abstract sebuah subject atau topic string. String query ini akan di eksekusi oleh perintah `$store->query` sesuai dengan string query. Untuk *endpoint* berisi target sparql endpoint yang akan dilakukan query, pada potongan kode tersebut endpoint merujuk pada variabel `$uri` yang bersikan string alamat server endpoint yang akan dituju. Fungsi pada potongan kode tersebut penulis tambahkan pada file `show_detail.inc.php` pada aplikasi SLiMS.

```
function getAdditionalinfo($uri){
    $remote_store_endpoint = 'http://'.explode_resource($uri).'/sparql';
    /* configuration */
    $config = array(
        /* remote endpoint (gene database)*/
        'remote_store_endpoint' => $remote_store_endpoint,
    );
    /* instantiation */
    $store = ARC2::getRemoteStore($config);
    if (isset($_GET['key']) AND !empty($_GET['key'])) {
        $keywords = $dbs->escape_string(urldecode(ltrim($_GET['key'])));
    } else {
        $keywords = '';
    }
    $q = ' PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
        PREFIX foaf: <http://xmlns.com/foaf/0.1/>
        SELECT ?keterangan ?primaryInfo ?label
        WHERE {
            <'.$uri.'> dbpedia-owl:abstract ?keterangan.
            <'.$uri.'> rdfs:label ?label.
            OPTIONAL {<'.$uri.'>
foaf:isPrimaryTopicOf ?primaryInfo}.
        }
    ';
    $rows = $store->query($q, 'rows');
    foreach ($rows as $row) {
        $uri_info = substr($row['keterangan'],0,500)." ... <br>".
        "<img src='images/default/wiki-icon.png'>
<a href='".$row['primaryInfo']."' target='_blank'> ".$row['label']."</a>
(Wikipedia Link)";
    }
}
```

```

return $suri_info;
}

```

Gambar 4.33 Potongan kode menampilkan data dari dbpedia.org

Pada data detail bibliografi juga menampilkan data judul koleksi tambahan yang diambil dari eksternal resource VIAF.org, penulis menggunakan fungsi parsing data xml dari hasil URI *linked data* author pada *ekseternal resource* VIAF.org. Seperti yang terlihat pada potongan kode pada gambar 4.34 Untuk tautan pada judul koleksi pustaka alamat link menggunakan *instance seeAlso* dari data author.seeAlso yang kemudian digabungkan dengan string URI ke alamat pencarian koleksi pustaka di worldcat.org.

```

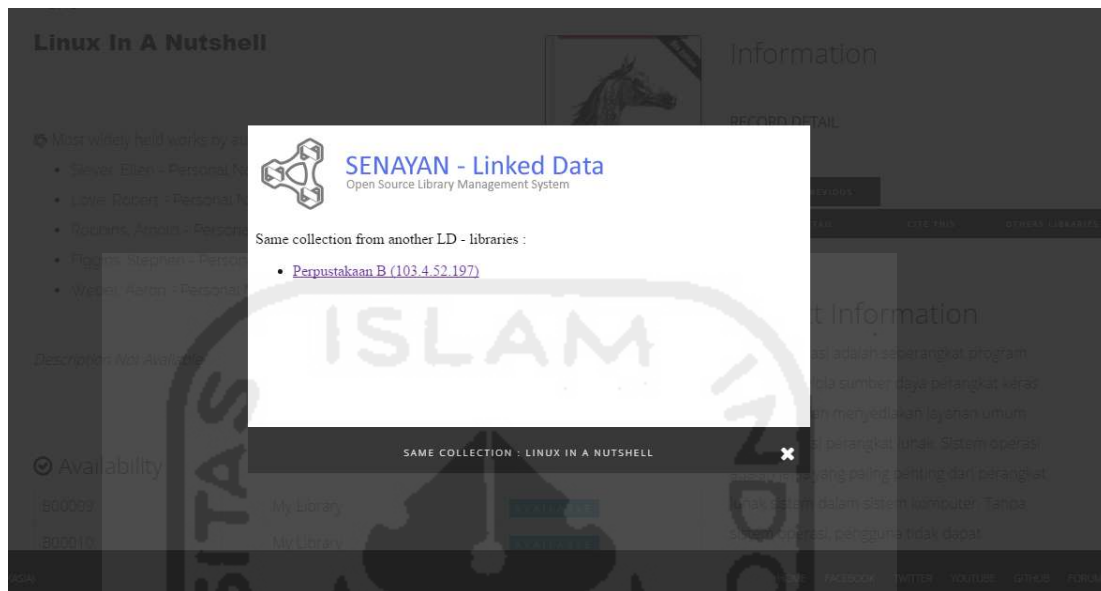
public function viaf_work($suri,$author) {
    $feed = file_get_contents($suri.'/viaf.xml');
    $sxe = new SimpleXMLElement($feed);
    $ns=$sxe->getNameSpaces(true);
    $ns2 = $sxe->children($ns['ns2']);
    $judul = "";
    foreach($ns2->titles as $data){
        foreach($data->work as $sdata){
            $title = htmlentities($sdata->title);
            $author = htmlentities($author);
            $link= "http://www.worldcat.org/search?q=ti%3A".
                $title."+au%3A".$author."&
                qt=advanced&dblast=638";
            $judul.= "- <a href='".$link."' target='_blank'>
                ".$sdata->title."</a><br>";
        }
    }
    return $judul;
}

```

Gambar 4.34 Potongan kode menampilkan data dari viaf.org

Pada modul menampilkan data koleksi juga ditambahkan fungsi untuk menampilkan data koleksi yang sama pada perpustakaan lain yang juga menerapkan aplikasi SLiMS dengan menggunakan teknologi *linked data* seperti yang terlihat

pada gambar 4.35. Untuk bisa mentautkan data yang sama pada perpustakaan lain dibutuhkan endpoint string yang berfungsi sebagai alamat dari sumber data lain.



Gambar 4.35 Tampilan data koleksi yang sama pada perpustakaan lain

Adapun potongan kode fungsi untuk menampilkan data koleksi sejenis dari perpustakaan lain adalah seperti yang terlihat pada gambar 4.36

```
function getKoleksiPerpusLainIsbn($endpoint,$slims_uri,$key){
    global $sysconf;
    $remote_store_endpoint = $endpoint;
    /* configuration */
    $config = array(
        /* remote endpoint (gene database)*/
        'remote_store_endpoint' => $remote_store_endpoint,
    );
    /* instantiation */
    $store = ARC2::getRemoteStore($config);
    if (isset($_GET['key']) AND !empty($_GET['key'])) {
        $q = 'PREFIX schema: <http://schema.org/>
            PREFIX dc: <http://purl.org/dc/elements/1.1/>
            PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
            PREFIX db: <http://103.4.52.197:8181/slims8/resource/>
            PREFIX d2r: <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-
server/config.rdf#>
            PREFIX owl: <http://www.w3.org/2002/07/owl#>
            PREFIX map: <http://103.4.52.197:8181/slims8/resource/#>
            PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX slims: <'$.slims_uri.'>
SELECT DISTINCT * WHERE {
  ?s rdf:type schema:Book.
  ?s schema:isbn "'. $key.'".
  ?s schema:name ?name.
  ?s slims:biblio_id ?id.
}
LIMIT 10

';
$rows = $store->query($q, 'rows');
foreach ($rows as $row) {
  $suri_info = $row['id'];
}
return $suri_info;
}

```

Gambar 4.36 Kode fungsi untuk menampilkan data koleksi dari perpustakaan lain

4.2.2.2. Modul penambahan URI *linked data*

Modul penambahan URI *linked data* terdiri dari dua modul yaitu modul penambahan data URI *linked data* author dan penambahan data URI *linked data* subject. Kedua modul tersebut terletak pada halaman administrator dimana user admin harus melakukan proses login terlebih dahulu untuk bisa melakukan proses penambahan URI *linked data* author dan subject. Dalam hal ini proses login penulis tidak melakukan pembuatan atau perubahan karena sudah tersedia pada aplikasi SLiMS sebelumnya.

Modul penambahan URI *linked data* yang pertama adalah modul data master author. Modul ini memiliki fungsi untuk menambah dan merubah data author atau penulis sebuah bibliografi. Pengembangan yang dilakukan adalah dengan menambahkan fungsi teknologi *linked data eksternal resource* lain pada data author baik yang sudah ada maupun data yang akan ditambahkan. Data author akan dikaitkan dengan eksternal resource dari VIAF.org seperti yang terlihat pada gambar 4.37 Pada proses ini sistem akan melakukan proses pencarian data author pada sistem VIAF.org, jika informasi author yang dimaksud telah ditemukan maka URI dari VIAF.org akan disimpan pada pada sistem SLiMS.

The screenshot displays the 'AUTHOR' management page. On the left is a dark sidebar with navigation items: ADMIN, SHORTCUT, DASHBOARD, OPAC, BIBLIOGRAPHY, CIRCULATION, MEMBERSHIP, MASTER FILE, STOCK TAKE, SYSTEM, and REPORTING. The main content area has a header with a menu icon, the title 'AUTHOR', and a user status 'You are currently logged in as Admin'. Below the header is a search bar with a 'SEARCH' button and two buttons: 'AUTHOR LIST' and '+ ADD NEW AUTHOR'. A green 'SAVE' button is positioned above the form. The form contains the following fields:

- Eksternal Resource**: A dropdown menu with 'viaf.org' selected.
- Author Name***: A text input field containing 'Teduh Dirgahaya', with a yellow highlight. To its right are two buttons: 'GET URI' (red) and 'CLEAR URI' (green).
- Search Result (URI)**: A list showing a search result: 'http://viaf.org/viaf/291574336 - Teduh Dirgahaya'.
- Author Birth Year**: An empty text input field.
- Authority Type**: A dropdown menu with 'Personal Name' selected.
- Authority Files**: An empty text input field.

At the bottom right of the form area, there is a dark button labeled 'CHAT WITH MEMBERS'.

Gambar 4.37 Halaman untuk menambahkan URI *linked data* author

Potongan kode proses penambahan *linked data* author seperti terlihat pada gambar 4.38, proses dimulai dengan mengisikan nama author, secara otomatis sistem akan melakukan proses pencarian data pada sistem VIAF.org. Hasil pencarian akan ditampilkan dalam bentuk *data list* yang memungkinkan user admin dapat memilih data author yang sesuai. Selain itu disediakan tombol link “view data” untuk melihat data author secara lengkap pada web VIAF.org seperti yang ditampilkan pada gambar 4.39. Jika informasi data *linked data* author telah sesuai, user admin bisa melakukan proses simpan data, data yang disimpan adalah URI dari resource VIAF.org.

```

define('INDEX_AUTH', '1');
    require_once '../sysconfig.inc.php';
    // session checking
    require SB.'admin/default/session.inc.php';
    require SB.'admin/default/session_check.inc.php';
    $key = $_POST['key'];
    $assoc = false;

    $url = 'http://viaf.org/viaf/AutoSuggest?query='.str_replace(' ', '%20',
, strtolower($key));

    $client = curl_init($url);
    curl_setopt($client, CURLOPT_RETURNTRANSFER, 1);
    $response = curl_exec($client);
    curl_close($client);
    $id_viaf = "";
    $link_viaf = "";
    $json1 = explode('[', $response);
    $hasilx = "";
    if(count($json1) > 1){
        $json2 = explode(']', $json1[1]);
        $json3 = "[".$json2[0]."]";
        $r1_viaf = json_decode($json3, true);
        for($x=0;$x<count($r1_viaf);$x++)
        {
            if($r1_viaf[$x]['viafid'] != $id_viaf){
                $id_viaf = $r1_viaf[$x]['viafid'];
                $link_viaf = "http://viaf.org/viaf/".$r1_viaf[$x]['viafid'];
                $data = $r1_viaf[$x]['term'].'~'.$link_viaf;
                $hasilx .= "<li
onClick=\"selectURI('".$data."');\">".$link_viaf."</li>";
            }
        }
    }else{
        $hasilx = "Data not found";
    }

    $hasil = "<ul>".$hasilx."</ul>";


```

Gambar 4.38 Potongan kode penambahan *linked data* author

VIAF
Virtual International Authority File

Search

Select Field: Select Index: Search Terms: **Search**



Rowling, J. K., 1965-
Rowling, J.K.
J. K. Rowling British novelist
Rowling, Joanne Kathleen, 1965-....
Rowling, Joanne K.
Rowling, J.K. (Joanne Kathleen), 1965-
Rowling, Joanne Kathleen
1965 - רולינג, ג'יי. קיי.
1965 - رولینگ, ج. ك.
Galbraith, Robert 1965-
Rowling, J. K. 1965 July 31- xA
1965 - رولیغ, ج. ك.
VIAF ID: 116796842 (Personal)
Permalink: <http://viaf.org/viaf/116796842>
ISNI: [0000 0001 2148 628X](http://www.isni.org/0000-0001-2148-628X)

Gambar 4.39 Halaman tautan pada viaf.org

Modul penambahan *linked data* selanjutnya adalah modul master data subject atau topic. Pada modul ini seperti yang sudah dijelaskan sebelumnya bahwa user admin dapat mengkaitkan data dengan data pada eksternal resource lainnya dengan menggunakan teknologi *linked data*. Data yang akan dikaitkan adalah data subject atau topic seperti yang ditunjukkan pada gambar 4.40 User admin dapat mengkaitkan data subject dengan memilih *eksternal resource* yang akan digunakan, ada dua dua pilihan *eksternal resource* yaitu id.dbpedia.org dan dbpedia.org kemudian dilanjutkan dengan memasukkan nama subject atau topic yang di inginkan. Sistem akan secara otomatis akan melakukan proses pencarian data pada eksternal resource yang telah dipilih sebelumnya. Jika data telah ditemukan akan di tampilkan pada kolom pilihan yang sekaligus akan menampilkan potongan informasi dari hasil pencarian yang dipilih.

The screenshot shows the 'SUBJECT' management page. The user is logged in as 'Admin'. The page has a search bar and a 'SEARCH' button. Below the search bar, there is a 'SAVE' button and a form for adding a subject. The 'Eksternal Resource' dropdown is set to 'id.dbpedia.org'. The 'Subject*' field contains 'Universitas Islam Indonesia', with 'GET URI' and 'CLEAR URI' buttons. Below this, a 'Search Result (JURI)' section displays a list of URIs for 'Universitas Islam Indonesia'. At the bottom, there is a 'Classification Code' field and a 'CHAT WITH MEMBERS' button.

Gambar 4.40 Halaman untuk menambahkan URI *linked data* subject

Cuplikan kode pada modul penambahan *linked data* subject tersebut dapat diperlihatkan pada gambar 4.41 data informasi subject atau topic yang ditampilkan merupakan hasil query ke *eksternal resource* (id.dbpedia.org atau dbpedia.org). Proses pencarian data di mulai dari penentuan endpoint string yang dilanjutkan dengan membuat sparql query string dan di akhiri dengan mengeksekusi query tersebut. Hasil dari query tersebut akan di tampilkan dalam format json dan ditampilkan dalam bentuk dropdown list sekaligus menampilkan data abstraksi informasi yang dihasilkan.

```

$resource = $_GET['r'];
switch($resource){
//id.dbpedia.org
case '1' : $remote_store_endpoint = 'http://id.dbpedia.org/sparql';
break;
case '2' : $remote_store_endpoint = 'http://dbpedia.org/sparql';
break;
}
/* configuration */
$config = array(
'remote_store_endpoint' => $remote_store_endpoint,
);
/* instantiation */
$store = ARC2::getRemoteStore($config);

```

```

if (isset($_GET['key']) AND !empty($_GET['key'])) {
    $keywords = $dbs->escape_string(urldecode(ltrim($_GET['key'])));
} else {
    $keywords = '';
}
switch($resource){
//id.dbpedia.org
case '1' : $q = 'PREFIX dbpedia-owl:
    <http://dbpedia.org/ontology/>
    select ?a ?b ?c ?direct ?noDirect ?labelDirect
    where {
    ?a rdfs:label "'.ucwords($keywords).'"@id.
    ?a rdfs:label ?c.
    OPTIONAL {?a dbpedia-owl:wikiPageRedirects ?b}.
    OPTIONAL {?a dbpedia-owl:abstract ?noDirect}.
    OPTIONAL {?b dbpedia-owl:abstract ?direct}.
    OPTIONAL {?b rdfs:label ?labelDirect}.
    }
    limit 20';
    break;

//dbpedia.org
case '2' : $q = 'PREFIX dbo: <http://dbpedia.org/ontology/>
    select ?a ?b ?pageRedirect $abstrak
    where {
    ?a rdfs:label "'.ucwords($keywords).'"@en.
    ?a rdfs:label ?b.
    OPTIONAL {?a dbo:wikiPageRedirects ?pageRedirect}.
    OPTIONAL {?pageRedirect
    <http://dbpedia.org/ontology/abstract> ?abstrak}.
    FILTER (langMatches(lang(?abstrak),"en")).
    }';
    break;
}
$rows = $store->query($q, 'rows');
if($errs = $store->getErrors()){
    echo '<option value="0">x'.$errs[0].'\</option>';
}

$hasil = '{"query": "'.ucwords($keywords).'", "result": [';
$j = count($rows);
foreach ($rows as $row) {
switch($resource){

```

```

//id.dbpedia.org
case '1' : $uri = (!empty($row['b'])) ? $row['b'] : $row['a'];
//label direct URI
$label = (!empty($row['labelDirect'])) ?
        $row['labelDirect'] : $row['c'];
//abstrak direct uri
if(!empty($row['direct'])){
    $dAbstract = trim($row['direct']);
}elseif(!empty($row['noDirect'])){
    $dAbstract = trim($row['noDirect']);
}else{
    $dAbstract = "-";
}
$primaryTopic = "https://id.wikipedia.org/wiki/".
                $label;
break;
case '2' : $uri = (!empty($row['pageRedirect'])) ?
$row['pageRedirect'] : $row['a'];
$label = $row['b'];
if(!empty($row['pageRedirect'])){
    $dAbstract = trim($row['abstrak']);
}else{
    $dAbstract = "-";
}
$primaryTopic = "https://id.wikipedia.org/wiki/".
                $label;
break;
}

// show image
$labelImg = str_replace(' ', '_', $label);
$urlImage = "https://id.wikipedia.org/w/api.php?action=query&
            titles=".$labelImg."&prop=pageimages&format=json";
$resultImage = curl($urlImage);
$decode = json_decode($resultImage, true);
$image_pages = $decode["query"]["pages"];
foreach($image_pages as $k1=>$v1){
    if(count($image_pages[$k1])> 3){
        $img = $image_pages[$k1]['thumbnail']['source'];
        $img2 = resizeImage($img, '100');
    }else{
        $img2 = "";
    }
}

```

```
}  
$data = $label.'~'. $uri.'~'. $img2.'~'. $resource.'~'. $primaryTopic.  
      '~'. addslashes($dAbstract);  
$hasilx .= "<li onClick=\"selectURI('".$data."');\">". $uri."</li>";  
}  
$hasil = "<ul>". $hasilx."</ul>";
```

Gambar 4.41 Potongan kode proses menambahkan URI *linked data* subject

