



RANCANG BANGUN SISTEM REPOSITORI BERBASIS WEB SEMANTIK MENGUNAKAN ONTOLOGI

Prianda Dipati Bangsa¹, Indra Hermawan²

¹Program Studi Teknik Informatika

²Program Studi Teknik Multimedia dan Jaringan

Jurusan Teknik Informatika dan Komputer

Politeknik Negeri Jakarta

Depok, Jawa Barat, Indonesia

priandadipati@gmail.com, indra.hermawan@tik.pnj.ac.id

Abstract

A repository is a system used to store data in the form of information such as archival documents, journals, or papers in an institutional institution. In this research, a repository system design will be made for the Department of Informatics and Computers at the Jakarta State Polytechnic based on the semantic web. Semantic technology allows the system to understand the meaning of information. This repository uses apache jena fuseki as data storage with SPARQL, which is made with the Resource Description Framework scheme. The final result of this research becomes a repository system that has been successfully applied to the final project collection in 2020 with regular and specific search features to facilitate the search for a document.

Keywords: *Repository, semantic web, ontology, RDF, SPARQL*

Abstrak

Repositori adalah sebuah sistem yang digunakan untuk melakukan penyimpanan data berupa informasi seperti dokumen kearsipan, jurnal, atau karya tulis dalam sebuah lembaga institusi. Dalam penelitian kali ini akan dibuat perancangan sistem repositori untuk jurusan Teknik Informatika dan Komputer pada institusi Politeknik Negeri Jakarta dengan berbasiskan web semantik. Teknologi semantik memungkinkan sistem dapat mengerti suatu makna pada suatu informasi. Pembuatan repositori ini menggunakan apache jena fuseki sebagai penyimpanan data dengan SPARQL yang dibuat dengan skema *Resource Description Framework*. Hasil akhir dari penelitian ini menjadi sebuah sistem repositori dimana telah berhasil diterapkan pada pengumpulan tugas akhir pada tahun 2020 ini dengan terdapat fitur pencarian regular dan juga spesifik dalam mempermudah pencarian suatu dokumen.

Kata kunci: *Repositori, web semantik, ontologi, Resource Description Framework, SPARQL*

1. PENDAHULUAN

Umumnya arsip elektronik digunakan untuk membuat duplikasi data atau informasi dari sebuah catatan arsip yang telah dibuat. Salah satu lembaga yang menggunakan arsip elektronik ini adalah pada perguruan tinggi yang dikenal dengan nama Repositori Institusi Perguruan Tinggi atau sering disebut Repositori Institusi (RI). Penggunaan Repositori Institusi bersifat *Open Access* (OA) yang bertujuan untuk penyimpanan informasi terkait dengan penelitian seperti jurnal, tugas akhir (TA), praktik kerja lapangan (PKL), tesis, dan disertasi. Dengan melakukan penyimpanan data secara digital tersebut maka kemudahan mempublikasikan informasi kepada

masyarakat umum juga semakin mudah guna dapat dilakukan untuk penelitian lebih lanjut.

Pembuatan repositori memiliki dua model rancangan berdasarkan teknologinya. Rancangan pertama adalah repositori era web 2.0 yaitu dengan memanfaatkan metadata dan juga *protocol* sebagai pengelompokkan dan penyebaran informasi. Rancangan kedua adalah repositori era web 3.0 yaitu dengan memanfaatkan metode semantik dimana *website* repositori dapat mengerti makna dari suatu kalimat yang ada pada sistem Politeknik Negeri Jakarta (PNJ) khususnya pada Jurusan Teknik Informatika dan Komputer (TIK) saat ini belum memiliki sistem repositori yang dapat menampung dokumen laporan kegiatan. Sehingga sangat menyulitkan dalam penyimpanan dan

pengelolaan dokumen laporan kegiatan tersebut. Berdasarkan hal tersebut penulis mengambil langkah untuk mengajukan judul penelitian ini dalam pembuatan sistem repositori untuk jurusan TIK. Setelah melakukan analisis kebutuhan melalui wawancara yang didapatkan dari ibu Risna Sari melalui pesan whatsapp dan email, maka didapatkan beberapa kebutuhan sistem yaitu (1) pembuatan sistem repositori ini ditujukan untuk penyimpanan dokumen seperti skripsi, laporan PKL, SK-SK, dokumen jadwal perkuliahan, dan surat-surat. (2) pengguna repositori adalah akademisi dari jurusan TIK (dosen, staff, dan mahasiswa). (3) Batasan akses kepada *user* seperti *upload* dan *download* dibuat dalam beberapa skenario.

Pembangunan repositori tersebut akan digunakan penerapan teknologi web semantik dimana sistem dapat memahami makna setiap dokumen yang telah diarsipkan pada sistem. Perancangan web semantik melibatkan ontologi yang memiliki entitas pada setiap sumber informasi. Kemudian entitas-entitas tersebut akan saling terhubung satu dengan yang lainnya yang akan membentuk suatu relasi. Semakin banyak entitas dalam suatu informasi maka akan semakin mudah juga komputer dalam memahami suatu makna pada informasi tersebut. Sehingga dari proses tersebut akan membentuk suatu rancangan sistem berbasis semantik. Dengan demikian pembuatan web repositori berbasis semantik ini diharapkan dapat membantu jurusan TIK dalam mengelola arsip yang masih disimpan secara konvensional kedalam bentuk digital dan juga membantu menyebarkan informasi penelitian secara digital kepada civitas akademik di internal maupun eksternal.

2. TINJAUAN PUSTAKA

2.1 Arsip

Arsip adalah media cetak seperti buku, dokumen, naskah, dan juga lainnya yang memiliki informasi penting dari suatu kegiatan tertentu pada suatu organisasi atau institusi [2]. Sedangkan kearsipan adalah suatu kegiatan dalam melakukan penyimpanan dokumen pada suatu tempat tertentu agar dokumen dapat mudah dikelola dengan baik demi menjaga keaslian dokumen tersebut [5]. Dengan demikian maka pengertian arsip adalah catatan berisi rekam jejak informasi atau peristiwa dengan bentuk tertulis berisikan huruf, angka, maupun gambar yang memiliki maksud untuk memberikan informasi mengenai suatu pelaksanaan kegiatan yang telah berlalu.

2.2 Web Semantik

Web semantik adalah teknologi *website* yang memungkinkan *website* dapat memahami suatu makna dari informasi [6]. *Website* yang dirancang dengan teknologi semantik memiliki arsitektur tersendiri. Berikut ini adalah komponen dari arsitektur web semantik:

- a. **URI** atau *Uniform Resource Identifier* adalah suatu alamat atau identitas yang digunakan untuk menghubungkan masing-masing sumber data.
- b. **Unicode** adalah teknologi untuk pengkodean dan merepresentasikan teks pada suatu informasi.
- c. **XML + NS + xmlschema** adalah suatu format yang digunakan dalam pertukaran informasi pada *website*. Format tersebut berupa *markup* tanpa *tag* khusus dan tidak bersinggungan dengan HTML.
- d. **Resource Description Framework (RDF) + rdf schema** adalah suatu format data dalam pembentukan ontologi. Rdfschema merupakan suatu skema dari kumpulan entitas yang terhubung satu dengan lainnya sehingga membentuk sebuah makna.
- e. **Ontology vocabulary** adalah suatu bahasa yang digunakan untuk membentuk suatu entitas pada informasi. Entitas tersebut nantinya akan saling berkaitan dimana semakin banyak entitas pada suatu informasi maka akan semakin mudah sistem dalam mengerti suatu makna tersebut.
- f. **Logic dan proof** adalah memastikan suatu sumber informasi dapat dibuat ontologi sesuai dengan standar yang telah ditetapkan.
- g. **Trust** adalah proses untuk memastikan bahwa sumber informasi yang dihasilkan bisa dipercaya dan dipertanggung jawabkan. Oleh karena itu proses ini didukung oleh digital *signature* untuk menjaga keaslian informasi.

2.3 Repositori

Repositori adalah suatu sistem penyimpanan dokumen berupa video, gambar, maupun teks secara elektronik dengan tujuan mempermudah penyimpanan data. Repositori biasa digunakan pada institusi dan juga lembaga pemerintahan dalam menyimpan dokumen penting ataupun yang bersifat rahasia. Dengan semakin berkembangnya teknologi, saat ini banyak repositori yang sudah beralih dalam bentuk digital. Repositori digital adalah penyimpanan dokumen penting elektronik dengan tujuan memperluas penyebaran informasi dokumen agar informasi dapat digunakan untuk kepentingan lebih lanjut [3].

2.4 Ontologi

Ontologi adalah bagian dari ilmu filsafat yang menjelaskan mengenai keadaan sesuatu secara meluas. Menurut istilah Ontologi adalah ilmu hakekat yang menyelidiki alam nyata ini dan bagaimana keadaan yang sebenarnya [4]. Dalam ilmu komputer ontologi memiliki definisi yaitu "*An ontology is an explicit specification of a*" yang artinya adalah sebuah ontologi adalah sebuah spesifikasi yang eksplisit dari suatu konseptualisasi. Sehingga pada dasarnya ontologi digunakan untuk memperluas suatu makna sehingga menjadi keterkaitan antara satu dengan yang lainnya.

Ontologi memiliki beberapa manfaat jika diterapkan pada sistem yaitu:

- a. Menjelaskan informasi secara eksplisit. Artinya, jika terdapat suatu informasi yang dijelaskan dengan ontologi maka informasi tersebut dapat diketahui “hulu” dan “hilir”nya sehingga pemahaman makna pada suatu informasi bisa dapat disimpulkan dengan baik.
- b. Penggunaan ulang domain pada ontologi. Dengan ini maka makna dari suatu ontologi akan bisa berkembang luas.

Oleh karena itu penggunaan ontologi yang sebelumnya dapat digunakan kembali dan juga dapat dikembangkan sesuai dengan pengarahannya. Ontologi memiliki beberapa elemen yaitu:

a. Instance

Elemen instance adalah menyatakan suatu objek pada ontologi. Elemen ini biasanya menyatakan suatu nama tempat, orang, hewan, tumbuhan, dan juga lainnya.

b. Property

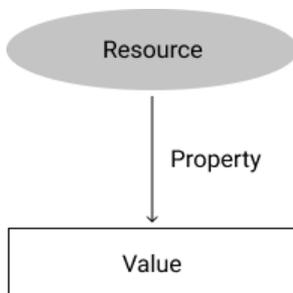
Property merupakan panah pendukung antara instance. Dengan adanya property suatu informasi yang terhubung dapat menjadi sebuah makna. Property yang dihubungkan terdapat suatu value untuk bisa memberikan informasi antara hubungan objek satu dengan objek lainnya.

c. Class

Class merupakan himpunan yang berisi objek. Di dalam class pasti terdapat lebih dari satu objek sehingga banyaknya objek dalam satu class bisa direpresentasikan dengan mudah. Contoh di dalam class hewan terdapat tiga objek yaitu kucing, ayam, dan sapi. Maka dengan demikian: “kucing sub Class Of hewan”, “ayam sub Class Of hewan”, dan “sapi sub Class Of hewan”.

2.5 RDF

RDF adalah format data yang digunakan untuk membentuk konsep triple dalam bentuk subjek predikat dan objek pada sebuah informasi. RDF juga dikenal dengan standar untuk mendeskripsikan suatu data. Konsep RDF mirip denan ERD tetapi dengan kegunaan yang berbeda. RDF digunakan dalam pembuatan web semantik sementara ERD dilakukan dalam membuat basis data relasional.



Gambar 1. RDF Triple

Pada Gambar 1 RDF memiliki tiga elemen dalam merepresentasikan data yaitu resource, property dan value. Resource adalah elemen untuk memberitahu informasi apa yang ingin dijelaskan. Property merupakan penghubung

antara resource dan value yang memiliki arti lain sebagai kata sambung antara subjek dan objek. Value merupakan data yang akan dijelaskan dari suatu resource.

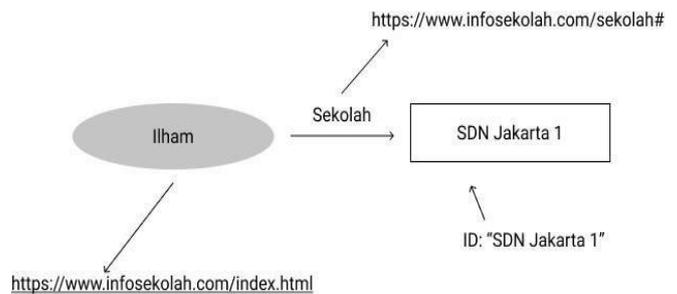


Gambar 2. Contoh RDF Triple

Pada Gambar 2 terdapat contoh sebuah sumber data berbentuk RDF. Kata “Ilham” dapat digambarkan sebagai resource yang artinya sumber RDF tersebut akan menjelaskan mengenai orang bernama “Ilham”. Kata “Sekolah” dianggap sebagai property yang artinya ingin menjelaskan “Ilham Sekolah dimana”. Sedangkan “SDN Jakarta 1” adalah direpresentasikan sebagai value dari “Sekolah Ilham”.

Bentuk triple pada RDF tidak dapat membalik hubungan antara subjek dan objek. Sehingga urutan dalam menafsirkan konsep triple hanya bisa dimulai dari kiri ke kanan, atau dari subjek ke objek.

1. RDF naming adalah merepresentasikan sebuah informasi RDF dengan tujuan agar komputer dapat memahami sebuah informasi tersebut.



Gambar 3. RDF Naming

```

<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#" xmlns:eric="http://www.w3.org/People/EH/contact#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
<rdf:Description rdf:about="http://www.w3.org/People/EH/contact#me"
  <contact:fullName>Eric Miller</contact:fullName>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/People/EH/contact#me"
  <contact:mailbox rdf:resource="mailto:e.miller123@tjexample"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/People/EH/contact#me"
  <contact:personalTitle>Dr.</contact:personalTitle>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/People/EH/contact#me"
  <rdf:type rdf:resource="http://www.w3.org/2000/10/swap/pim/contact#Person"/>
</rdf:Description>
</rdf:RDF>
  
```

Gambar 4. RDF Syntax

Alamat URI yang tertulis pada Gambar 3 merupakan sebuah ilustrasi. Maka jika diakses halaman tidak dapat ditemukan. Penulisan RDF naming berbentuk URI yang direpresentasikan dalam sebuah format RDF. Pada element resource dan property RDF naming harus berupa URI. Sedangkan pada element value RDF naming boleh

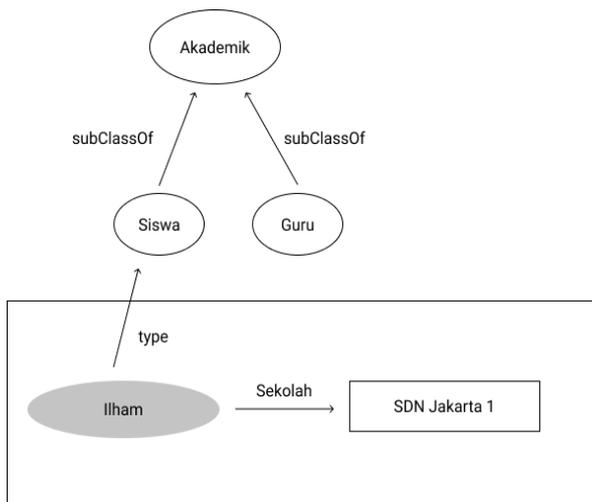
direpresentasikan dengan URI ataupun *value* seperti diatas. Pada *element property* URI tidak akan mengembalikan dalam bentuk halaman. Artinya URI tersebut hanya sebagai identitas sebuah Id dengan keterangan *property* tersebut.

RDF *Syntax* adalah format penulisan pada model RDF. Terdapat beberapa macam bentuk *syntax* seperti RDF/XML, *N-Triple*, N3, Turtle, dan lainnya. Format RDF/XML adalah yang sering digunakan. Gambar 4 adalah contoh RDF/XML:

2.6 RDFS (RDF Schema)

RDFS atau RDF Schema adalah perluasan dari model RDF yang akan membentuk taksonomi pada desain web semantik. RDFS digunakan untuk merepresentasikan suatu informasi pada RDF. Dengan kata lain, kosakata yang ada pada RDF akan memiliki entitas dalam maksud memperluas sebuah makna. Dalam RDFS, taksonomi yang ada akan membentuk makna secara umum sehingga akan membentuk sebuah *graph*.

Pada Gambar 5 kosakata “Ilham” memiliki type dari “Siswa”. Dimana “Siswa” memiliki adalah *sub Class Of* dari “Akademik”. Sehingga hal tersebut dapat disimpulkan bahwa “Ilham” adalah bagian dari “Siswa”. Sedangkan kata “Siswa” memiliki makna yaitu seorang akademisi yang diartikan dari sebuah kosa kata “Akademik”.



Gambar 5. RDF Shema

2.7 Ontology Web Language (OWL)

Ontology Web Language (OWL) adalah sebuah bagian pada pembuatan web semantik yang digunakan untuk merepresentasikan informasi dalam bentuk *file* yang kemudian akan diimplementasikan pada pengembangan *website* dalam penggunaan semantik dalam memahami informasi pada suatu makna.[1]. OWL Memiliki beberapa elemen diantaranya:

a. Class

Class digunakan untuk pendefinisian suatu rancangan pada objek. Dengan adanya *class* maka setiap objek memiliki

class tersendiri sesuai dengan penempatan makna dalam sebuah *ontology* [1].

b. Individual

Individual atau disebut juga *instance* adalah anggota (*member*) dari kelas. *Instance* merupakan objek dari pada suatu *class* yang nantinya dapat dihubungkan oleh dengan individual lainnya [1].

c. Property

Property adalah relasi antara instance dan juga RDF. instance memiliki beberapa contoh seperti *sub Property Of* yang digunakan untuk batasan suatu domain [1].

2.8 SPARQL

SPARQL adalah perintah atau bahasa yang digunakan untuk mengakses *query* pada sebuah model data semantik pada format data RDF. SPARQL bisa disebut juga sebagai bahasa untuk mengakses *linked data* dengan penggunaan *end point* untuk dapat menghasilkan relasi antara satu informasi dengan yang lainnya. Bahasa SPARQL dianggap setara dengan SQL dengan memiliki format sintak yang serupa, hanya saja dalam penggunaannya berbeda.

SPARQL memiliki dua macam *syntax* yang digunakan untuk mengakses *query*. Pertama adalah *read oriented query*, yaitu untuk menampilkan data pada *server*. Berikut ini adalah sintak pada *read oriented query* SPARQL:

SELECT QUERY digunakan untuk mengembalikan data yang telah diseleksi untuk ditampilkan dengan parameter yang ditentukan.

CONSTRUCT QUERY digunakan untuk mengembalikan data dalam format RDF yaitu menampilkan data berupa *triple* SPO.

ASK QUERY digunakan untuk mengembalikan data untuk dibandingkan apakah kira-kira pola *query* tersebut sesuai atau tidak.

DESCRIBE QUERY digunakan untuk mengembalikan seluruh informasi data yang terkait untuk ditampilkan.

Berikut ini adalah sintak *write oriented query* SPARQL:

CREATEQUERY digunakan untuk membuat baru *graph empty*.

INSERT QUERY digunakan untuk memasukkan data baru ke dalam dokumen RDF dengan adanya parameter yang telah ditentukan.

COPY QUERY digunakan untuk menduplikasi kalimat pada model data RDF ke dalam rancangan *graph*.

ADD QUERY digunakan untuk menambahkan kalimat pada model data RDF ke dalam rancangan *graph*.

MOVE QUERY digunakan untuk memindahkan kalimat pada model data RDF dari satu dokumen ke dalam dokumen *graph* lain.

DELETE QUERY digunakan untuk menghapus kalimat pada model data RDF pada dokumen yang telah diseleksi

2.9 Laravel

Laravel adalah *framework* berbasis PHP dalam memudahkan pembuatan *website*. Laravel memiliki konsep *Model-View-Controller* (MVC) sehingga memudahkan manajemen antara *component* seperti *database* dan juga *layout* [7].

3. METODE PENELITIAN

Metode pada penelitian yang digunakan dalam pembuatan sistem ini adalah *Software Development Life Cycle* (SDLC). SDLC adalah suatu proses dalam pembuatan sistem atau aplikasi dengan menggunakan beberapa tahap atau metode yang telah teruji agar rancangan sistem yang digunakan mudah dipahami [8]. Berikut ini merupakan tahapan dari siklus SDLC:

3.1 Analisa Kebutuhan

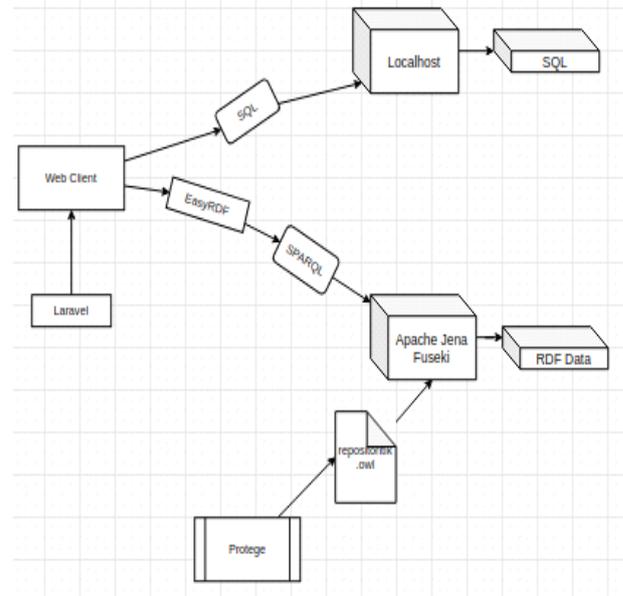
Proses ini merupakan awal dalam melihat alur kerja pada sistem yang ingin dibuat. Dari mulai pengidentifikasian masalah hingga mengetahui kebutuhan *user* atau pengguna.

3.2 Perancangan / Desain

Perancangan atau Desain adalah tahap dimana rancangan sistem digambarkan seperti merancang alur *database* diagram, merancang UI, hingga mendesain alur kerja sistem. Tahap ini melanjutkan proses sebelumnya yaitu Analisis Kebutuhan. Selain itu, pada tahap ini dilakukan juga rancangan arsitektur sistem.

Disini penulis melakukan analisa terhadap pembuatan repositori. Maka didapatkan rancangan seperti Gambar 6. Pertama, sistem repositori menggunakan *framework* laravel yang kemudian disambungkan pada dua *database* yaitu SQL dan SPARQL. Untuk menyambungkan *database* menggunakan SQL dapat dilakukan langsung melalui *framework* laravel sedangkan untuk SPARQL terdapat library EasyRDF.

Selanjutnya repositori disambungkan kepada dua *host* yaitu localhost dan Apache Jena Fuseki. Apache Jena Fuseki dapat menampung *file* berformat owl yang dihasilkan dari Protege yaitu *software* yang digunakan untuk merancang skema RDF.



Gambar 6. Perancangan Sistem

3.3 Penerapan / Implementasi

Penerapan atau Implementasi adalah tahap dimana melanjutkan proses perancangan / desain dimana implementasi akan memulai tahap pembuatan sistem seperti menuliskan *code program*.

3.4 Integrasi dan Testing

Integrasi dan Testing adalah tahap akhir dimana semua proses sebelumnya sudah dilakukan. Tahap ini merupakan tahap pengujian sistem untuk memastikan semua proses yang telah dilakukan bisa berjalan dengan yang direncanakan. Dalam perusahaan maupun institusi penyimpanan dokumen sudah seharusnya dibuat

4. HASIL DAN PEMBAHASAN

Sistem yang telah selesai di buat dibutuhkan pengujian untuk mengetahui apakah rancangan sistem telah sesuai dengan requirement yang telah diberikan dan apakah sistem telah berjalan dengan sempurna. Berikut ini adalah beberapa pengujian yang dilakukan pada repositori:

4.1 Pengujian *Black Box*

Black box testing adalah pengujian yang befokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi *input* dan melakukan pengetesan pada fungsional program [9][10]. Berdasarkan dari hasil pengujian sebanyak 43 skenario tercatat memiliki keberhasilan sebesar 100%.

Tabel 1. Pengujian *Black Box*

Item Uji	Butir Uji	Jenis Pengujian
Registrasi	Melakukan pendaftaran akun	<i>Black Box</i>
Autentikasi	Melakukan login	<i>Black Box</i>
	Melakukan <i>logout</i>	<i>Black Box</i>

Item Uji	Butir Uji	Jenis Pengujian
Menu Beranda Admin	Halaman beranda <i>admin</i>	<i>Black Box</i>
Menu Pengumuman Admin	Menampilkan data pengumuman	<i>Black Box</i>
	Halaman edit pengumuman	<i>Black Box</i>
Menu Dokumen Admin	Menampilkan halaman data dokumen	<i>Black Box</i>
	Halaman tambah dokumen	<i>Black Box</i>
	Halaman edit dokumen	<i>Black Box</i>
Menu Kategori Admin	Menampilkan halaman kategori	<i>Black Box</i>
	Halaman edit kategori	<i>Black Box</i>
Menu Sub Kategori	Halaman sub kategori	<i>Black Box</i>
	Halaman edit sub kategori	<i>Black Box</i>
Menu Profil Dosen	Menampilkan data pribadi user dosen yang sedang login	<i>Black Box</i>
	Halaman Edit profil data dosen	<i>Black Box</i>
Menu Dokumen Dosen	Menampilkan halaman data dokumen	<i>Black Box</i>
	Halaman tambah dokumen	<i>Black Box</i>
	Halaman edit dokumen	<i>Black Box</i>
Menu Profil Mahasiswa	Menampilkan data pribadi <i>user</i> dosen yang sedang <i>login</i>	<i>Black Box</i>
	Halaman Edit profil data dosen	<i>Black Box</i>
Menu Dokumen Dosen	Menampilkan halaman data dokumen	<i>Black Box</i>
	Halaman tambah dokumen	<i>Black Box</i>
	Halaman edit dokumen	<i>Black Box</i>

Tabel 1 terdapat item uji dimana masing-masing item memiliki sub butir uji. Masing-masing level *user* pastinya memiliki item uji registrasi dan autentikasi karena setiap *user* harus melakukan *login* dan *logout* pada sistem. Sistem repositori memiliki tiga level *user* yaitu admin, dosen, dan mahasiswa yang dapat mengakses repositori. Admin memiliki fitur lengkap dalam pengelolaan repositori seperti

pengelolaan *user* kategori, dan dokumen *User* dosen dan mahasiswa hanya dapat melakukan pengelolaan dokumen seperti tambah, edit, dan hapus dokumen.

4.2 Pengujian *User Acceptance Test*

User Acceptance Testing (UAT) merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah staff/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya [10]. Berdasarkan dari hasil pengujian sebanyak 3 yang dilakukan uji coba kepada empat mahasiswa, dua dosen, dan dua admin dengan memperoleh data seperti yang ditunjukkan pada Tabel 2, Tabel 3, dan Tabel 4..

Tabel 2. UAT *User* Mahasiswa

Pengujian UAT Pada <i>User</i> Mahasiswa			
No	Pertanyaan	Ya	Tidak
1.	Apakah fungsi <i>login</i> dapat digunakan	4	0
2.	Apakah fungsi Registrasi dapat digunakan dengan baik?	3	1
3.	Apakah fungsi edit profil bisa berjalan dengan baik?	4	0
4.	Apakah fungsi tambah dokumen dapat berjalan dengan baik?	4	0
5.	Apakah fungsi edit dokumen dapat berjalan dengan baik?	4	0
6.	Apakah fungsi <i>delete</i> dokumen dapat berjalan dengan baik?	4	0
7.	Apakah fungsi <i>download file</i> pada halaman spesifik Dokumen bisa berjalan dengan baik?	4	0
8.	Apakah fungsi pencarian dokumen dapat berjalan dengan baik?	4	0
Presentase		96.875% Menjawab Ya	

Tabel 3. UAT *User* Dosen

Pengujian UAT Pada <i>User</i> Dosen			
No	Pertanyaan	Ya	Tidak
1.	Apakah fungsi <i>login</i> dapat digunakan dengan baik?	1	0
2.	Apakah fungsi edit profil bisa berjalan dengan baik?	1	0
3.	Apakah Fungsi tambah dokumen dapat berjalan	1	0

Pengujian UAT Pada <i>User Dosen</i>			
No	Pertanyaan	Ya	Tidak
	dengan baik?		
4.	Apakah fungsi edit dokumen dapat berjalan dengan baik?	1	0
5.	Apakah fungsi <i>delete</i> dokumen dapat berjalan dengan baik?	1	0
6.	Apakah fungsi <i>download file</i> pada halaman spesifik dokumen bisa berjalan dengan baik?	1	0
7.	Apakah fungsi pencarian dokumen dapat berjalan dengan baik?	1	0
Presentase		100%	

Tabel 4. UAT *User Admin*

Pengujian UAT Pada <i>User Admin</i>			
No	Pertanyaan	Ya	Tidak
1.	Apakah fungsi <i>login</i> dapat digunakan	2	0
2.	Apakah fitur edit data pengumuman dapat berjalan dengan baik?	2	1
3.	Apakah fitur tambah kategori dapat berjalan dengan baik?	2	0
4.	Apakah fitur tambah sub kategori dapat berjalan dengan baik?	2	0
5.	Apakah fitur edit kategori dapat berjalan dengan baik?	2	0
6.	Apakah fitur edit sub kategori dapat berjalan dengan baik?	2	0
7.	Apakah fitur <i>delete</i> kategori dapat berjalan dengan baik?	2	0
8.	Apakah fitur <i>delete</i> sub kategori dapat berjalan dengan baik?	2	0
9.	Apakah fitur tambah admin dapat berjalan dengan baik?	2	0
10.	Apakah fitur tambah dosen dapat berjalan dengan baik?	2	0
11.	Apakah fitur tambah mahasiswa dapat berjalan dengan baik?	2	0
12.	Apakah fitur edit admin dapat berjalan dengan baik?	2	0

Pengujian UAT Pada <i>User Admin</i>			
No	Pertanyaan	Ya	Tidak
13.	Apakah fitur edit dosen dapat berjalan dengan baik?	2	0
14.	Apakah fitur edit mahasiswa dapat berjalan dengan baik?	2	0
15.	Apakah fitur <i>delete</i> admin dapat berjalan dengan baik?	2	0
16.	Apakah fitur <i>delete</i> dosen dapat berjalan dengan baik?	2	0
17.	Apakah fitur <i>delete</i> mahasiswa dapat berjalan dengan baik?	2	0
18.	Apakah fitur tambah data dokumen dapat berjalan dengan baik?	2	0
19.	Apakah fitur edit data dokumen dapat berjalan dengan baik?	2	0
20.	Apakah fitur <i>delete</i> data dokumen dapat berjalan dengan baik?	2	0
21.	Apakah fitur <i>download</i> data dokumen dapat berjalan dengan baik?	2	0
22.	Apakah fitur pencarian data dokumen dapat berjalan dengan baik?	1	1
Presentase		97.727% Menjawab Ya	

Hasil pada Tabel 2, Tabel 3, dan Tabel 4 setelah dilakukannya uji coba kepada *user* seperti mahasiswa, dosen, dan admin. Pengambilan sampel uji coba dilakukan dengan memberikan uji coba langsung sistem kepada mahasiswa dan dosen pada jurusan TIK. Hasilnya *user* admin memiliki persentase sebesar 97.727% dalam menjawab “Ya”. Mahasiswa memiliki persentase sebesar 96.875% dalam menjawab “Ya”. Dosen memiliki persentase sebesar 100% dalam menjawab “Ya”.

5. KESIMPULAN

Tujuan dari penelitian ini adalah untuk mengimplementasikan sistem repositori desain web semantik jurusan Informatika dan Teknik Komputer di Politeknik Negeri Jakarta. Setelah melakukan studi literatur, desain ontologi, desain sistem, dan juga implementasi. Maka dapat disimpulkan bahwa aplikasi ini sudah dapat digunakan untuk menyimpan data arsip jurusan TIK. Ini juga dapat didukung dalam penggunaan semantik untuk menampilkan data dalam pencarian dokumen. Ada dua jenis pencarian dokumen, yaitu:

- Pencarian biasa adalah fungsi untuk mencari dokumen menggunakan satu kata kunci. Kata kunci ini akan dicocokkan dengan data yang berada di server

menggunakan kueri SQL. Jika pencocokan kata kunci akurat maka sistem akan menampilkan dokumen terkait sesuai dengan kata kunci.

- b. Pencarian spesifik adalah pencarian kata kunci dokumen tertentu. Pencarian akan melibatkan konsep semantik bagian tiga. Pencarian dilakukan dengan menampilkan banyak pilihan kolom seperti kategori, sub kategori, pencarian karakter dengan memasukkan pengguna dan memilih berdasarkan kotak kolom yang tersedia. Dari setiap pencarian akan dimasukkan ke dalam kueri SPARQL dalam proses pencarian data.

Berdasarkan pengerjaan sistem repositori yang dibuat dengan *framework* Laravel dengan berbasis semantik tentunya masih banyak kekurangan. Maka dari itu penulis memberikan saran untuk pengembangan lebih lanjut, yaitu:

- a. Sistem dikembangkan dengan menggunakan RESTful API dalam pengolahan data ke server. Tujuannya agar arsitektur dapat dikelola lebih mudah.
- b. Penggunaan *database* PostgreSQL agar penampungan data dapat lebih maksimal

DAFTAR PUSTAKA

- [1] Gunawan, Fandi Halim, Penerapan Web Semantik Untuk Aplikasi Pencarian Pada Repositori Koleksi Penelitian, Studi Kasus: Program Studi Sistem Informasi STIKOM MIKROSKIL MEDAN
- [2] Hasugian, J. (2003). Pengantar kearsipan.
- [3] Armstrong, Michelle, 2013. Institutional repository management models that support faculty research dissemination.
- [4] Anwar, M. (2015). *Filsafat pendidikan*. Kencana.

- [5] A.Sayuti dan Kurniawati. 2013. Manajemen Sarana dan Prasarana di SMK 1 Kasihan Bantul. *Jurnal Akuntabilitas Manajemen Pendidikan*, Vol. 1, No. (1): 98- 108.
- [6] Wibisono, Riyan (2018). Aplikasi *Virtual Customer Service* Berbasis Semantik Menggunakan Ontologi. Undergraduate (S1) thesis, University of Muhammadiyah Malang.
- [7] Aminudin, 2015, Cara Efektif Belajar Framework Laravel, Lokomedia, Yogyakarta
- [8] S, Rosa A. Dan M.Shalahuddin. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- [9] Mustaqbal, M. Sidi, Roero Fajri Firdus, Hendra Rahmadi 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan* Vol.1, No.3
- [10] Perry, William E. 2006. *Effective Methods for Software Testing* 3rd Edition. Wiley Publishing, Inc. Indianapolis, Indiana