

# Pemanfaatan SPARQL Dalam Memperoleh Knowledge dari Database Framework Media Sosial Elgg yang Bermodel RDF *Friend of a Friend*

Carina Natalia<sup>1</sup>, Bernard R. Suteja, S.Kom., M.Kom.<sup>2</sup>

Teknik Informatika, Universitas Kristen Maranatha  
Jl. Prof. drg. Surya Sumantri No. 65 Bandung

<sup>1</sup>carinanatalia97@yahoo.com

<sup>2</sup>bernard.rs@it.maranatha.edu

**Abstract** — *With social media that implements RDF Friend of a Friend, the information data needed can be arranged regularly based on RDF provisions and can be used for information that can be produced effectively with the implementation of the RDF. One of the RDF model used is TTL that containing The FoaF which is arranged based on group data and contains prefixes that support the identity of each subject, predicate or object. The testing process will also compare between SPARQL and Database Queries that have a function that is displaying data*

**Keywords**— *RDF Friend of a Friend, Semantic Web, SPARQL, TTL, Query Database*

## I. PENDAHULUAN

Informasi dalam internet seiring perkembangan teknologi bertambah semakin banyak. Salah satunya adalah media sosial, yang seiring penerapan perkembangan teknologi dan memudahkan pengguna dalam berkomunikasi dan bersosialisasi secara *online*. Untuk membangun sebuah media sosial sebagai perawalan, solusi Framework Elgg secara default telah menyediakan beberapa fitur umum yang dibutuhkan sebuah sosial media, seperti blogging, file sharing, networking, groups, dan lain-lain. Selain fitur yang disediakan dari Framework Elgg, para developer juga telah banyak berkontribusi dalam pembuatan plugin-plugin yang bertujuan untuk meningkatkan performa dan keefisienan framework media sosial ini. Aktivitas yang terjadi dalam media sosial oleh berbagai user, dapat berkontribusi dalam sejumlah data yang dapat digunakan pihak developer untuk mengetahui informasi yang dapat disimpulkan. W3C (World Wide Web consortium) mengeluarkan konsep Web Semantik sebagai database global yang saling terhubung, yaitu dengan penggunaan teknik yang memberi kerangka umum yang memungkinkan data yang akan dibagi dan digunakan kembali seluruh aplikasi, perusahaan, dan batas-batas komunitas.

Salah satu komponen dari web semantik yang akan dimanfaatkan adalah RDF (*Resource Description Framework*) yang berfungsi sebagai model data sederhana yang berhubungan dengan sumber objek dan bagaimana cara mereka berhubungan. Teknik ini dapat diterapkan dalam pencarian informasi terkait sumber informasi (yaitu media sosial dalam penelitian ini). RDF *Friend of a Friend* merupakan salah satu meta data yang mendeskripsikan seseorang atau lebih dan jaringan sosial mereka secara semantik. Dengan media sosial yang mengimplementasi RDF Friend of a Friend (Web Semantik), dapat mengetahui ikatan pertemanan yang telah terjalin dalam media sosial seseorang (mutual friend), sehingga dari ikatan mutual friend tersebut diharapkan dapat terjalin hubungan yang baik/komunitas antar mahasiswa/i serta mengetahui informasi yang terkait dengan perjalinan mutual friend tersebut (misalnya persamaan pada jurusan, tahun angkatan, dll) dan penyaluran informasi dapat berdasarkan informasi mutual friend tersebut untuk mengakses lebih banyak mahasiswa, sehingga memperluas informasi yang dapat diperoleh oleh pihak developer.

Misalnya pengambilan data user berdasarkan syntax database (query) berdasarkan kondisi tertentu, seperti pengambilan nama user yang saling berteman. Namun hal ini memerlukan waktu kerja lebih banyak dibandingkan dengan teknik RDF yang menggunakan SPARQL dalam proses pengambilan data tersebut, karena proses data dari database telah dikonversi ke dalam bentuk teks RDF yang tersusun secara berkelompok, yang kemudian dapat diambil secara efektif dalam proses SPARQL.

## II. KAJIAN TEORI

Teori yang dijelaskan meliputi : *Framework* Elgg sebagai kerangka media sosial, Web Semantik untuk *knowledge base* media sosial, *The Friend of a Friend* dan *Menyimpulkan Fakta Menggunakan SPARQL*.

### A. *Framework Elgg sebagai kerangka media sosial*

Elgg adalah sebuah platform untuk social network yang berbasis open source dan dikeluarkan di bawah GNU (General Public License). Elgg menggunakan teknologi web server Apache, database MySQL dan PHP sebagai bahasa untuk scripting [1]. Elgg adalah kerangka(framework) pengembangan cepat dengan fitur sosial built-in. Ini sangat cocok untuk membangun aplikasi apa pun tempat pengguna masuk dan membagikan informasi; “Elgg (pronunciation) is a rapid development Framework with built-in social features. It is a great fit for building any app where users log in and share information” [2].

### B. *Web Semantik untuk knowledge base media sosial*

Web semantik adalah sebuah web yang mampu untuk mendeskripsikan sesuatu dalam sebuah cara yang komputer dapat mengerti. Web semantik bukan suatu web terpisah, tetapi sebuah ekstensi dari yang sudah ada, dimana informasi yang diberikan mempunyai pengertian definisi yang baik, lebih baik untuk memungkinkan komputer dan manusia dalam bekerja sama.

Untuk mencapai tujuannya dibutuhkan pemberian meaning ke dalam masing-masing konten (sebagai atribut) yang akan digunakan oleh teknologi web semantik, yaitu ke dalam beberapa layer sebagai berikut [3]:

1. XML Layer, untuk merepresentasikan data.
2. URI (Uniform Resource Identifier), adalah sebuah pengidentifikasi yang terdiri urutan karakter dengan aturan-aturan tertentu yang digunakan untuk mengakses suatu sumber daya.
3. RDF Layer, untuk merepresentasikan pengertian dari data.

XML(Extensible Markup Language) dikembangkan oleh W3C yang mempunyai tujuan untuk lebih menyempurnakan pada teknologi HTML yang telah menjadi dasar layanan berbasis web sekarang ini, dalam fungsinya XML memiliki dua fungsi yaitu sebagai format dokumen dan format pertukaran data pada sebuah sistem [7]. XML adalah bahasa markup yang menggunakan tag sebagai penanda untuk mengkategorikan, menjelaskan data lebih spesifik. Pada XML terdapat tag pembuka <tag> dan tag penutup </tag>. Tag pembuka dan tag penutup hanya mempunyai perbedaan karakter / pada tag penutup [5]. URI (Uniform Resource Identifier) sebagai pengidentifikasi unik yang menjadi pembeda antar resource [6]. RDF(Resource Description Framework) merupakan suatu metadata yang digunakan untuk mendeskripsikan alamat sumber daya pada web. Metadata ini dapat berupa judul, pengarang, hak cipta, dan lisensi dalam dokumen web [7]. Dokumen Tuttle (TTL) memungkinkan penulisan grafik RDF dalam bentuk tekstual yang ringkas. Sebuah grafik RDF terdiri dari tiga unsur utama, yaitu subjek, predikat dan objek.

```
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<#green-goblin>
  rel:enemyOf <#spiderman> ;
  a foaf:Person ;    # in the context of the Marvel universe
  foaf:name "Green Goblin" .

<#spiderman>
  rel:enemyOf <#green-goblin> ;
  a foaf:Person ;
  foaf:name "Spiderman", "Человек-паук"@ru .
```

Gambar 1. Contoh penulisan Turtle (TTL) [8]

Gambar 1 adalah contoh mengenai fitur dalam sebuah TTL, yaitu @base and Relative IRIs, daftar @prefix and prefixed names, predicate yang dipisahkan oleh tanda ';', daftar objek yang dipisahkan oleh ',', the token a (menandakan predikat @prefix rdf), and literals(objek) [8].

### C. The Friend of a Friend

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<#JW>
  a foaf:Person ;
  foaf:name "Jimmy Wales" ;
  foaf:mbox <mailto:jwales@bomis.com> ;
  foaf:homepage <http://www.jimmywales.com> ;
  foaf:nick "Jimbo" ;
  foaf:depiction <http://www.jimmywales.com/aus_img_small.jpg> ;
  foaf:interest <http://www.wikimedia.org> ;
  foaf:knows [
    a foaf:Person ;
    foaf:name "Angela Beesley"
  ] .

<http://www.wikimedia.org>
  rdfs:label "Wikimedia" .
```

Gambar 2. Contoh FOAF dalam Turtle(TTL) atau Notation-3 (N3) [9]

Profil FOAF pada Gambar 2 (ditulis dalam format Turtle) menyatakan bahwa Jimmy Wales (JW) adalah nama orang yang sedang dijelaskan. Alamat e-mail, homepage dan penggambarannya adalah sumber daya web, yang berarti masing-masing dapat digambarkan menggunakan RDF juga. Dia memiliki Wikimedia sebagai minat, dan mengenal Angela Beesley (yang merupakan nama sumber daya 'Person' dari foaf. FOAF (Friend Of A Friend) seperti web, yaitu sistem informasi yang terhubung, dibangun dengan teknologi web semantik yang desentralisasi dan dirancang untuk integrasi data antara beragam aplikasi, website dan service, serta sistem software. FOAF dirancang untuk dapat digunakan bersama seperti kamus lainnya (Schema atau Ontologi), dan dapat digunakan bersama-sama dengan berbagai macam tools umum dan layanan yang diciptakan untuk semantic web [10].

### D. Menyimpulkan Fakta Menggunakan SPARQL

SPARQL adalah bahasa yang digunakan berupa query untuk mendapatkan informasi dari suatu graph RDF [11]. SPARQL dapat melakukan pemrosesan informasi yang terdapat dalam sebuah domain (ontologi) sesuai dengan kebutuhan [12].

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

Gambar 3. Contoh Prefix yang digunakan dalam SPARQL [13]

```
SELECT ?name ?person ?description WHERE {
  ?person rdf:type dbo:Person.
  ?person foaf:name ?name .
  ?person rdfs:comment ?description .
  FILTER (regex(?name, 'morgan freeman', 'i')) .
  FILTER (lang(?description) = 'en')
} ORDER BY ?name
```

Gambar 4. Contoh Query SPARQL [13]

Sebagai contoh pengguna memasukan kata kunci “morgan freeman” dan memilih kategori *Person*, *Kode Program query* yang digunakan ditunjukkan pada Gambar 4. Pada *query* tersebut setidaknya menggunakan empat *prefix* yang ditunjukkan pada Gambar 3 [13].

### III. ANALISIS DAN RANCANGAN SISTEM

Pada framework media sosial elgg yang telah dijalankan, akan dimanfaatkan data aktivitas dari basis datanya untuk digunakan dalam proses pencarian knowledge dengan teknik RDF dengan dukungan Apache Jena sesuai dengan alur Gambar 5 berikut.



Gambar 5. Alur Rancangan

#### A. Alur Pencarian Knowledge

Berikut adalah langkah-langkah dari proses awal informasi dalam bentuk database hingga menjadi knowledge yang menunjukkan fakta informasi dalam media sosial ini.

1. *Konversi Database ke XML* : Tujuan dari konversi database ke XML dikarenakan teknik web semantik akan membaca data xml yang telah merepresentasikan data (tersusun) pada database yang akan dibuat knowledge untuk media sosial ini.

guid	name	username
36	Admin	admin
93	George	1472099
45	Dira	dira
46	Risega	risega
75	Carina Natalia	1572022
76	Ziavaniwati	1572012
77	Rachmi Rachmadiany	1572052

Gambar 6. Data User

Pada Gambar 6 merupakan tabel user dalam database framework elgg dengan nama “*elgg\_users\_entity*”, yang memiliki 3 kolom utama untuk menunjukkan identitas user, yaitu kolom “*guid*” sebagai ID user yang bersifat primary key, kolom “*name*” sebagai nama user yang ditampilkan pada media sosial dan kolom “*username*” sebagai identitas user yang dimanfaatkan untuk login.

id	type	subtype	action_type	access_id	view	subject_guid	object_guid
1	user		friend	2	river/relationship/friend/create	46	44
2	user		friend	2	river/relationship/friend/create	44	36
3	user		friend	2	river/relationship/friend/create	36	44
4	object	market	create	2	river/object/market/create	36	68
6	user		update	2	river/user/default/profileiconupdate	44	44
7	user		friend	2	river/relationship/friend/create	76	77
8	user		friend	2	river/relationship/friend/create	77	76
9	user		friend	2	river/relationship/friend/create	76	75
10	user		friend	2	river/relationship/friend/create	75	76

Gambar 7. Data Friendship

Tabel *Friendship* pada Gambar 7 merupakan *database framework* elgg yang bernama “*elgg\_river*” dengan berfungsi menyimpan aktivitas user, salah satunya yaitu aktivitas pertemanan antar user, yaitu view (sebagai informasi aktivitas), *subject\_guid*(sebagai user 1) dan *object\_guid*(sebagai user 2).

2. *XML hasil konversi dari Database* : Masing-masing data user dari data *Database* tersebut akan dikelompokkan berdasarkan nama user yang memiliki data sendiri dan nama-nama teman yang dimiliki yang tersusun menjadi sebuah bentuk data XML, seperti pada Gambar 8 berikut.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<user>
  <script type="text/javascript" charset="utf-8" id="zm-extension"/>
  <Mahasiswa>
    <nipnpr>1572022</nipnpr>
    <name>CARINA NATALIA</name>
    <hasFriend>ZIAVANIWATI</hasFriend>
    <hasFriend>NERISSA HANSA WIJAYA</hasFriend>
  </Mahasiswa>
  <Mahasiswa>
    <nipnpr>1572012</nipnpr>
    <name>ZIAVANIWATI</name>
    <hasFriend>CARINA NATALIA</hasFriend>
  </Mahasiswa>
  <Mahasiswa>
    <nipnpr>1572052</nipnpr>
    <name>RACHMI RACHMADIANY</name>
  </Mahasiswa>
  <Mahasiswa>
    <nipnpr>1572036</nipnpr>
    <name>NERISSA HANSA WIJAYA</name>
    <hasFriend>CARINA NATALIA</hasFriend>
  </Mahasiswa>
</user>

```

Gambar 8. XML hasil konversi dari Database

3. *Konversi XML menjadi TTL* :

```
1 @prefix user: <http://socialzone.hosting.it.maranatha.org/profile/>.
2 @prefix role: <http://socialzone.hosting.it.maranatha.org/role/>.
3 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
4
5 user:1572022 a role:Mahasiswa;
6     user:hasName "CARINANATALIA";
7     foaf:hasFriend user:1572012 ;
8     foaf:hasFriend user:1572036 .
9
10 user:1572012 a role:Mahasiswa;
11     user:hasName "ZIAVANIWATI";
12     foaf:hasFriend user:1572022 .
13
14 user:1572052 a role:Mahasiswa;
15     user:hasName "RACHMI RACHMADIANY";
16     user:hasRole "Mahasiswa".
17
18 user:1572036 a role:Mahasiswa;
19     user:hasName "NERISSA HANSA WIJAYA";
20     foaf:hasFriend user:1572022 .
21
```

Gambar 9. Hasil Konversi XML menjadi TTL

Gambar 9 merupakan TTL hasil konversi XML menggunakan 1 prefix user (URI identifikasi) dan 2 prefix default W3. Data user disusun berdasarkan kelompok dan memiliki relasi pertemanan sendiri dan akan didaftarkan sebagai foaf untuk setiap user.

#### IV. IMPLEMENTASI

Proses implementasi terbagi menjadi 3 tahap sesuai dengan rancangan. Tahap pertama yaitu pemanfaatan PHP untuk proses konversi database ke bentuk data XML.

```
<?php
header("Content-type: text/xml");
$koneksi = mysqli_connect("localhost","root","", "1572022_media_sosial");

//Query pengambilan data yang dibutuhkan
$perintah="select * from elgg_users_entity";
$perintah2="SELECT u1.name, u1.username, u1.email, u2.name, u2.username, u2.email FROM elgg_river f
INNER JOIN elgg_users_entity u1 ON u1.guid=f.subject_guid
INNER JOIN elgg_users_entity u2 ON u2.guid=f.object_guid";
$hasil=mysqli_query($koneksi,$perintah);
$hasil2=mysqli_query($koneksi,$perintah2);
//Penulisan XML
echo "<user>";
while($baris=mysqli_fetch_row($hasil))
{
    //$user=$baris[3];
    $user="Mahasiswa";
    echo "<". $user. ">";
    echo "<nipnrp>$baris[2]</nipnrp>";
    echo "<name>$baris[1]</name>";
    echo "<email>$baris[6]</email>";
    mysqli_data_seek($hasil2,0);
    while($baris2=mysqli_fetch_row($hasil2))
    {
        if($baris[1]==$baris2[3] || $baris2[3]==$baris[1]){
            echo "<hasFriend>$baris2[2]</hasFriend>";
        }
    }
    echo "</". $user. ">";
}
echo "</user>";
?>
```

Gambar 10. Kode Program PHP Konversi Database to XML

Hasil dari konversi pada Gambar 10 ini yaitu mendaftarkan data-data user ke dalam bentuk XML(Gambar 8) dengan syarat menampilkan nama teman bila user tersebut pernah melakukan penambahan teman (sesuai pada tabel *friendship*). Proses konversi *database* ke XML diawali dengan *import database* dengan menggunakan NIP. Data yang akan dipilih adalah data kedua user yang telah berteman, berupa nama, *username* dan *email*.

```
<?xml version="1.0" encoding="UTF-8" ?>
<user>
  <script type="text/javascript" charset="utf-8" id="zm-extension"/>
  <Mahasiswa>
    <nipnrp>admin</nipnrp>
    <name>Admin</name>
    <email>carinanatalia97@gmail.com</email>
  </Mahasiswa>
  <Mahasiswa>...</Mahasiswa>
  <Mahasiswa>...</Mahasiswa>
  <Mahasiswa>...</Mahasiswa>
  <Mahasiswa>
    <nipnrp>1572012</nipnrp>
    <name>Ziavaniwati</name>
    <email>zia@gmail.com</email>
    <hasFriend>rachmi@gmail.com</hasFriend>
    <hasFriend>zia@gmail.com</hasFriend>
  </Mahasiswa>
  <Mahasiswa>
    <nipnrp>1572052</nipnrp>
    <name>Rachmi Rachmadiany</name>
    <email>rachmi@gmail.com</email>
    <hasFriend>rachmi@gmail.com</hasFriend>
    <hasFriend>zia@gmail.com</hasFriend>
  </Mahasiswa>
  <Mahasiswa>
    <nipnrp>1572022</nipnrp>
    <name>Carina Natalia</name>
    <email>carinanatalia97@yahoo.com</email>
  </Mahasiswa>
  <Mahasiswa>
    <nipnrp>1572036</nipnrp>
    <name>Nerissa Hansa Wijaya</name>
    <email>hansa@gmail.com</email>
  </Mahasiswa>
</user>
```

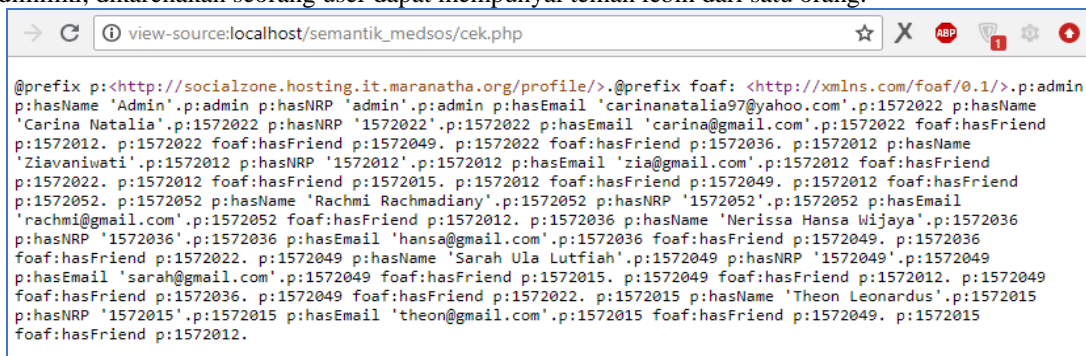
Gambar 11. Hasil XML

Gambar 11 merupakan hasil XML yang digenerate dari database media sosial. Data yang dipilih berupa “guid”, “name”, “friend relationship” dan “email”. Hasil eksekusi dilakukan dalam koneksi lokal yaitu “localhost/[direktori file PHP]”, dengan syarat file PHP berada dalam direktori “C:\xampp\htdocs\[direktori file PHP]”.

```
<?php
//Import XML
$xml=simplexml_load_file("http://localhost/semantik_medsos/xml.php") or die("Error: Cannot
create object");
//Daftar Prefix TTL
echo "@prefix p:<http://socialzone.hosting.it.maranatha.org/profile/>.";
echo "@prefix foaf: <http://xmlns.com/foaf/0.1/>.";
//Penulisan TTL (bersumber dari XML)
foreach($xml->Mahasiswa as $user) {
    echo "p: "; echo $user->nipnrp . " "; echo "p:hasName ";
    echo "" $user->name . " ";
    echo "p: "; echo $user->nipnrp . " "; echo "p:hasNRP ";
    echo "" $user->nipnrp . " ";
    echo "p: "; echo $user->nipnrp . " "; echo "p:hasEmail ";
    echo "" $user->email . " ";
    if (($user->hasFriend != "" )){
        for ($x = 0; $x < ($user->hasFriend->count()); $x++) {
            echo "p: ";
            echo $user->nipnrp . " ";
            echo "foaf:hasFriend ";
            echo "p: " $user->hasFriend[$x] . " ";
        }
    }
}
```

Gambar 12. Kode Program PHP Konversi XML to TTL

Pada script PHP Gambar 12 ini akan dilakukan impor data XML yang telah dibuat sebelumnya (Gambar 10) dan pendaftaran prefix yang akan digunakan dalam TTL ini. Proses penulisan data user sesuai dengan data XML (Gambar 8) dengan penambahan variabel prefix untuk identitas node. Node “hasFriend” dilakukan proses looping berdasarkan jumlah teman yang dimiliki, dikarenakan seorang user dapat mempunyai teman lebih dari satu orang.



```
@prefix p:<http://socialzone.hosting.it.maranatha.org/profile/>.@prefix foaf: <http://xmlns.com/foaf/0.1/>.p:admin
p:hasName 'Admin'.p:admin p:hasNRP 'admin'.p:admin p:hasEmail 'carinanatalia97@yahoo.com'.p:1572022 p:hasName
'Carina Natalia'.p:1572022 p:hasNRP '1572022'.p:1572022 p:hasEmail 'carina@gmail.com'.p:1572022 foaf:hasFriend
p:1572012. p:1572022 foaf:hasFriend p:1572049. p:1572022 foaf:hasFriend p:1572036. p:1572012 p:hasName
'Ziavaniwati'.p:1572012 p:hasNRP '1572012'.p:1572012 p:hasEmail 'zia@gmail.com'.p:1572012 foaf:hasFriend
p:1572022. p:1572012 foaf:hasFriend p:1572015. p:1572012 foaf:hasFriend p:1572049. p:1572012 foaf:hasFriend
p:1572052. p:1572052 p:hasName 'Rachmi Rachmadiany'.p:1572052 p:hasNRP '1572052'.p:1572052 p:hasEmail
'rachmi@gmail.com'.p:1572052 foaf:hasFriend p:1572012. p:1572036 p:hasName 'Nerissa Hansa Wijaya'.p:1572036
p:hasNRP '1572036'.p:1572036 p:hasEmail 'hansa@gmail.com'.p:1572036 foaf:hasFriend p:1572049. p:1572036
foaf:hasFriend p:1572022. p:1572049 p:hasName 'Sarah Ula Lutfiah'.p:1572049 p:hasNRP '1572049'.p:1572049
p:hasEmail 'sarah@gmail.com'.p:1572049 foaf:hasFriend p:1572015. p:1572049 foaf:hasFriend p:1572012. p:1572049
foaf:hasFriend p:1572036. p:1572049 foaf:hasFriend p:1572022. p:1572015 p:hasName 'Theon Leonardus'.p:1572015
p:hasNRP '1572015'.p:1572015 p:hasEmail 'theon@gmail.com'.p:1572015 foaf:hasFriend p:1572049. p:1572015
foaf:hasFriend p:1572012.
```

Gambar 13. Hasil Konversi dari XML ke TTL

Gambar 13 adalah TTL hasil konversi XML disusun berdasarkan kelompok pada XML (data setiap user). Hasil eksekusi dilakukan dalam koneksi lokal yaitu “localhost/[direktori file PHP]”, dengan syarat file PHP berada dalam direktori “C:\xampp\htdocs\[direktori file PHP]”.



Hasil TTL akan dilakukan proses akhir yaitu pemanfaatan SPARQL untuk menampilkan knowledge tertentu. Berikut adalah file .JSP yang akan mengeksekusi knowledge SPARQL.

```
<%@ page import="org.apache.jena.rdf.model.*" %>
<%@ page import="org.apache.jena.util.FileManager" %>
<%@ page import="java.io.*" %>
<%@ page import="org.apache.jena.rdf.vocabulary.*" %>
<%@ page import="org.apache.jena.query.*" %>
<%@ page import="org.apache.jena.query.ResultSetFormatter" %>
<%@ page trimDirectiveWhitespaces="true" %>

<%
//Import TTL
String inputFile = "http://localhost/semantik_medsos/cek.php";
Model model = ModelFactory.createDefaultModel();
InputStream in = FileManager.get().open(inputFile);
if(in == null){
    found = "TTL";
}
model.read(in, null, "TTL");
String command = "prefix p: <http://socialzone.hosting.it.marannatha.org/profile/>
prefix foaf: <http://xmlns.com/foaf/0.1/> SELECT ?p WHERE (?person
foaf:hasFriend p:1572012 . ?person p:hasName ?p)";

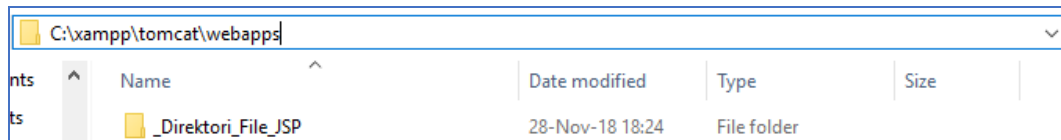
//String command1 = "prefix p:
<http://socialzone.hosting.it.marannatha.org/profile/> prefix foaf:
<http://xmlns.com/foaf/0.1/> select ?Angkatan_15 ?Nama where (?person
p:hasName ?Nama ?person p:hasNRP ?Angkatan_15. Filter (substr
(?Angkatan_15,1,2) = '15')";

//String command2 = "prefix p:
<http://socialzone.hosting.it.marannatha.org/profile/> prefix foaf:
<http://xmlns.com/foaf/0.1/> select ?Google_Mail ?Nama where (?person
p:hasName ?Nama ?person p:hasEmail ?Google_Mail Filter
(contains(?Google_Mail,'gmail.com') .)";

//Pilih salah satu command yang ingin ditampilkan
Query query = QueryFactory.create(command);
QueryExecution qe = QueryExecutionFactory.create(query,model);
ResultSet hasil = qe.execSelect();
String output = ResultSetFormatter.asText(hasil);
out.println(output);
%>
```

Gambar 14. Kode Program JSP SPARQL

Langkah awal yang dilakukan berupa melakukan impor terhadap library Apache Jena ke localhost Tomcat yang akan digunakan dalam proses eksekusi file JSP ini. File JSP yang berisi kode program seperti Gambar 14 disimpan pada direktori (Gambar 15) dan dijalankan melalui web browser dengan alamat "localhost:8080/\_Direktori\_File\_JSP/[Nama File JSP]".



Gambar 15. Direktori File JSP dalam Tomcat

A. Knowledge yang akan ditampilkan berdasarkan ketiga command tersebut adalah sebagai berikut

1. Knowledge "Friendship":

String command = "prefix p:

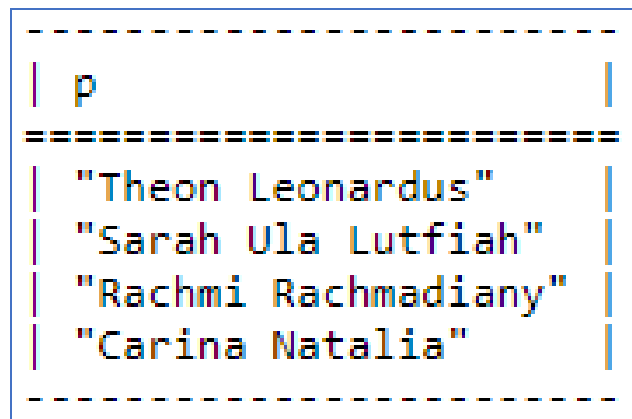
```
<http://socialzone.hosting.it.maranatha.org/profile/>
```

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?p WHERE {?person foaf:hasFriend p:1572012 . ?person
```

```
p:hasName ?p}";
```

Kode Program SPARQL "command" ini mencari nama user yang memiliki teman (F:hasFriend) dengan user beridentitas f:1572012 dan ditampilkan datanya ke kolom 'p'.



Gambar 16. Hasil SPARQL knowledge "Friendship"

Hasil SPARQL pada Gambar 16 menampilkan fakta tentang user "1572012" memiliki 4 orang teman yang bernama "Theon Leonardus", "Sarah Ula Lutfiah", "Rachmi Rachmadiany" dan "Carina Natalia" dalam media sosial ini.

2. Knowledge "Angkatan":

String command1 = "prefix p:

```
<http://socialzone.hosting.it.maranatha.org/profile/>
```

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
```

```
select ?Angkatan_15 ?Nama where {?person p:hasName ?Nama.?person
```

```
p:hasNRP ?Angkatan_15. Filter (substr (?Angkatan_15,1,2) = '15')}";
```

Kode Program SPARQL ini mencari tahun angkatan user berdasarkan NIPnya (asumsi 2 digit angka pertama NIP user menunjukkan tahun angkatannya sebagai mahasiswa).

Angkatan_15	Nama
"1572015"	"Theon Leonardus"
"1572049"	"Sarah Ula Lutfiah"
"1572036"	"Nerissa Hansa Wijaya"
"1572052"	"Rachmi Rachmadiany"
"1572012"	"Ziavaniwati"
"1572022"	"Carina Natalia"

Gambar 17. Hasil SPARQL *knowledge* "Angkatan"

Hasil SPARQL pada Gambar 17 menghasilkan fakta tentang tahun angkatan user dengan kolom "Angkatan\_15" berisi NIP user dan kolom "Nama" berisi nama user.

3. *Knowledge "Protokol Email"* :

String command2 = "prefix p:

```
<http://socialzone.hosting.it.maranatha.org/profile/>
```

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
```

```
select ?Google_Mail ?Nama where {?person p:hasName ?Nama.?person
```

```
p:hasEmail ?Google_Mail.Filter (contains(?Google_Mail,'gmail.com') ) .}";
```

Kode Program SPARQL ini mencari protokol email yang digunakan user berdasarkan email yang dimiliki user. Setiap protokol memiliki keunikan tersendiri seperti protokol *Gmail* tertulis @gmail.com, protokol *Yahoo Mail* tertulis @yahoo.com, dan lain-lain.

Google_Mail	Nama
"theon@gmail.com"	"Theon Leonardus"
"sarah@gmail.com"	"Sarah Ula Lutfiah"
"hansa@gmail.com"	"Nerissa Hansa Wijaya"
"rachmi@gmail.com"	"Rachmi Rachmadiany"
"zia@gmail.com"	"Ziavaniwati"
"carina@gmail.com"	"Carina Natalia"

Gambar 18. Hasil SPARQL *knowledge* "Protokol Email"

Hasil SPARQL pada Gambar 18 menghasilkan fakta tentang protokol tertentu yang digunakan user (Misalnya "theon@gmail.com" yang digunakan user "Theon Leonardus").

4. *Knowledge "Deskripsi user"* :

String command = " prefix p: <http://socialzone.hosting.it.maranatha.org/profile/>

```
prefix foaf: <http://xmlns.com/foaf/0.1/> describe ?person WHERE
```

```
{?person foaf:hasFriend p:1572012}";
```

Kode Program SPARQL ini adalah contoh menampilkan deskripsi data semua user yang memiliki teman dengan NIP "1572012" (terdiri dari 4 user berdasarkan Gambar 19) dalam bentuk model TTL.

```

@prefix p:      <http://socialzone.hosting.it.maranatha.org/profile/> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .

p:1572022 p:hasEmail    "carina@gmail.com" ;
          p:hasNRP      "1572022" ;
          p:hasName     "Carina Natalia" ;
          foaf:hasFriend p:1572012 , p:1572049 , p:1572036 .

p:1572052 p:hasEmail    "rachmi@gmail.com" ;
          p:hasNRP      "1572052" ;
          p:hasName     "Rachmi Rachmadiany" ;
          foaf:hasFriend p:1572012 .

p:1572049 p:hasEmail    "sarah@gmail.com" ;
          p:hasNRP      "1572049" ;
          p:hasName     "Sarah Ula Lutfiah" ;
          foaf:hasFriend p:1572015 , p:1572012 , p:1572036 , p:1572022 .

p:1572015 p:hasEmail    "theon@gmail.com" ;
          p:hasNRP      "1572015" ;
          p:hasName     "Theon Leonardus" ;
          foaf:hasFriend p:1572049 , p:1572012 .
    
```

Gambar 19. Hasil SPARQL knowledge Deskripsi user

Hasil SPARQL pada Gambar 4.18 menampilkan deskripsi lengkap dari setiap user yang memilih teman dengan identitas "1572012".

5. Model graph baru dengan fungsi "Construct" :

String command = " prefix p: <http://socialzone.hosting.it.maranatha.org/profile/>

prefix foaf: <http://xmlns.com/foaf/0.1/> construct {p:1572012 p:totalFriend ?count} where {select (count(?o) as ?count) where {p:1572012 foaf:hasFriend ?o}}";

Kode Program SPARQL ini membuat sebuah model graph baru dengan contoh knowledge berupa menampilkan total teman yang dimiliki oleh user "1572012" dengan predikat "p:totalFriend" dalam bentuk model TTL.

```

@prefix p:      <http://socialzone.hosting.it.maranatha.org/profile/> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .

p:1572012 p:totalFriend 4 .
    
```

Gambar 20. Model graph baru dengan fungsi "Construct"

Hasil SPARQL pada Gambar 4.19 menampilkan deskripsi lengkap dari setiap user yang memilih teman dengan identitas "1572012".

## V. PENGUJIAN

Pengujian ini akan menampilkan perbedaan serta tujuan penggunaan SPARQL dibandingkan Query Database dalam pengambilan data. Data yang menjadi objek perbandingan yaitu menampilkan user yang saling berteman. Jika SPARQL valid, maka akan menampilkan user yang saling berteman, sebaliknya jika Query Database valid, maka akan menampilkan user yang saling berteman.

Langkah Pengujian :

1. Membuat SPARQL untuk menampilkan user yang saling berteman.  
SPARQL = "prefix p: <http://socialzone.hosting.it.maranatha.org/profile/>  
prefix foaf: <http://xmlns.com/foaf/0.1/> select ?user ?teman where {?user foaf:hasFriend ?teman}"
2. Membuat Query Database untuk menampilkan user yang saling berteman.  
Query = "SELECT u1.name, u2.name FROM elgg\_river f  
INNER JOIN elgg\_users\_entity u1 ON u1.guid=f.subject\_guid  
INNER JOIN elgg\_users\_entity u2 ON u2.guid=f.object\_guid";  
Hasil pengujian dapat dilihat pada tabel 5.9 berikut :

TABEL 1  
IMPLEMENTASI MODEL GRAPH BARU DENGAN FUNGSI "CONSTRUCT"

Kasus Uji	Hasil Harapan	Hasil Pengamatan	Simpulan
SPARQL valid	Menampilkan user yang saling berteman	Menampilkan user yang saling berteman	Sesuai Harapan
Query Database valid	Menampilkan user yang saling berteman	Menampilkan user yang saling berteman	Sesuai Harapan

Kesimpulan pengujian berupa Syntax pada SPARQL lebih sederhana dibandingkan Query database dalam pencarian data yang sama.

#### VI. KESIMPULAN

Kontribusi dari sebuah media sosial berupa sejumlah data yang dapat diolah berdasarkan aktivitas-aktivitas yang terjadi. Terutama dengan pemanfaatan RDF yang berperan sebagai model data yang saling berhubungan, teknik ini dapat diterapkan dalam pencarian informasi terkait sumber informasi yang didapatkan sebelumnya, misalnya contoh kasus Framework media sosial Elgg yang dirancang untuk Fakultas Teknologi Informasi Universitas Kristen Maranatha.

Pemanfaatan RDF dalam penelitian ini melewati tiga tahap, yaitu proses konversi database ke XML dengan tujuan penyaringan dan penyusunan data yang akan digunakan, kemudian dilakukan konversi dari XML ke TTL dengan tujuan penambahan prefix sebagai identitas setiap node user dan kemudian dilakukan proses pencarian knowledge (fakta yang dapat ditampilkan) dengan menggunakan SPARQL dan ditampilkan pada halaman web.

#### DAFTAR PUSTAKA

- [1] M. Tjahjadi, T. Wiradinata and A. Suryaputra, "IMPLEMENTASI MODUL JEJARING SOSIAL UNTUK PENGEMBANGAN BISNIS INDONESIA TRIPPLANNER," 1 Agustus 2010. [Accessed 21 Juni 2018].
- [2] "http://learn.Elgg.org/en/stable/index.html," [Online]. [Accessed 21 Juni 2018]
- [3] Gunawan and F. Halim, "PENERAPAN WEB SEMANTIK UNTUK APLIKASI Pencarian PADA REPOSITORI KOLEKSI PENELITIAN, STUDI KASUS: PROGRAM STUDI SISTEM INFORMASI STMIK MIKROSKIL MEDAN," April 2014. [Accessed 21 Juni 2018].
- [4] D. Cahyo Prakoso and Y. Prayudi, "Model Enkripsi XML Pada Output DFXML untuk Pengamanan Metadata Bukti Digital," *Jurnal Masyarakat Informatika (Jumanji)*, vol. 1, 2017. [Accessed 11 Oktober 2018].
- [5] Sudirman, "ANALISIS KOMUNIKASI DATA DENGAN XML DAN JSON PADA WEBSERVICE," *CESS (Journal Of Computer Engineering, System And Science)*, vol. 1, July 2016. [Accessed 5 Juli 2018]
- [6] A. Faizul Ikrom and A. Susanto, "IMPLEMENTASI REST WEB SERVICE PADA APLIKASI PENGOLAH PESAN YAHOO MESSENGER PADA CV. MELIANA PRATAMA," 2015. [Accessed 11 Oktober 2018].
- [7] A. Hariyadi Supriyatna and I. S. M. Bastian, "Aplikasi Pencarian Pariwisata Perairan Di Indonesia Dengan menggunakan pendekatan Web Semantik (2014)". [Accessed 21 Juni 2018].
- [8] "RDF 1.1 Turtle," *W3 Org*, 2014. [Accessed 14 Oktober 2018].
- [9] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/FOAF\\_\(ontology\)](https://en.wikipedia.org/wiki/FOAF_(ontology)). [Accessed 11 November 2018]. [Accessed 14 Oktober 2018].
- [10] W. S. M. Prof. Dr. I Simri Wicaksana, L. M. Dr. Yuniar B., L. S. M. Dr. Wulandari, D. S. M. Dr. Purnamasari and M. S. M. Andriansyah, "Ontologi Data Instansi Pemerintah". [Accessed 17 Juni 2018]
- [11] N. Syihabil Haq and K. Muslim L, "Penerapan Sparql Dan Ontology Pada Pencarian Data Buku Dan Perbandingannya Dengan Pendekatan Relasional," 2018. [Accessed 20 Juni 2018]
- [12] M. Yani, "Implementasi Teori Refinement pada Pengkomposisian Soal pada Game berbasis Web Semantik," 2016. [Accessed 5 Juli 2018]
- [13] E. Al Ma'arif, "MESIN Pencari MEMANFAATKAN DBPEDIA KNOWLEDGE BASE," *Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri 2018*, 2018. [Accessed 14 Oktober 2018].