

Rancang Bangun Aplikasi Visualisasi Silsilah Keluarga Berbasis Ontologi

Christian Candrabiantara, Daniel Oranova Siahaan, dan Umi Laili Yuhana

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

E-mail : daniel@its-sby.edu

Abstrak—Silsilah keluarga merupakan suatu bagan yang menampilkan susunan keluarga dalam suatu pohon. Silsilah keluarga memiliki informasi yang menyatakan hubungan antar anggota yang kompleks. Dalam bidang kedokteran, silsilah keluarga digunakan untuk membantu diagnosa suatu penyakit seperti penyakit turunan. Para ahli genealog dapat menggunakan silsilah keluarga untuk menemukan dan menganalisis fakta-fakta menarik tentang sejarah keluarga mereka, seperti pola penamaan, persaingan antar saudara, atau peristiwa penting lainnya. Silsilah keluarga yang terus berkembang dan semakin kompleks membuat pemodelan relasional semakin sulit. Pemodelan semantik dibutuhkan untuk menyelesaikan masalah ini. Artikel ini membangun aplikasi yang dapat mencari anggota keluarga berdasarkan *query* dari pengguna menggunakan pemodelan semantik, khususnya ontologi. Aplikasi ini menggunakan fitur *query by design* untuk menambahkan dan mencari data dalam basis data. Aplikasi ini berhasil memodelkan silsilah keluarga termasuk relasi yang khas di masyarakat Indonesia. *Query by design* dapat diimplementasi dengan baik.

Kata kunci—Ontologi, Pemodelan Data, Silsilah Keluarga.

I. PENDAHULUAN

MANUSIA memerlukan informasi mengenai silsilah keluarganya untuk berbagai hal. Silsilah keluarga berguna untuk mempererat ikatan batin antar anggota keluarga, mempermudah anak cucu dalam menelusuri asal usul keluarganya, penentuan dalam pewarisan dan perkawinan, dan untuk keperluan medis agar penyakit turunan mudah dideteksi [1].

Perangkat lunak untuk membuat silsilah keluarga dapat dibuat dengan berbagai metode diantaranya dengan menggunakan metode relasional dan ontologi. Metode relasional dilakukan dengan *query* yang kompleks. Sedangkan dengan menggunakan metode ontologi, *query* dapat dilakukan dengan lebih mudah dengan terlebih dahulu membuat pemodelan data dan *rule* [2].

FamilyFinder adalah perangkat lunak yang berfungsi sebagai *editor* untuk membuat dan mencari silsilah keluarga dengan berbasis ontologi. FamilyFinder memiliki sistem masukan dan pencarian anggota keluarga yang berbeda dari kebanyakan aplikasi lain, yaitu menggunakan *query by design*. FamilyFinder juga dapat memodelkan silsilah keluarga yang khas di Indonesia [3].



Gambar 1. Skema Triplet

II. URAIAN PENELITIAN

A. Pemodelan Data

Pemodelan data RDF hampir mirip dengan konsep pemodelan diagram kelas, yaitu dengan membuat *statement* tentang *resource* dalam bentuk subjek, predikat, dan objek yang dikenal sebagai triplet [4]. Pemodelan triplet dalam silsilah keluarga dapat digambarkan seperti pada Gambar 1.

Subjek dan objek menunjukkan *resource*. Predikat menunjukkan ciri-ciri atau aspek dari *resource* serta menyatakan hubungan antara subjek dan objek. Contoh pernyataan “Budi *is father of* Joni” dapat direpresentasikan “Budi” sebagai subjek. “*isFatherOf*” sebagai predikat. “Joni” sebagai objek.

B. RDF Graph

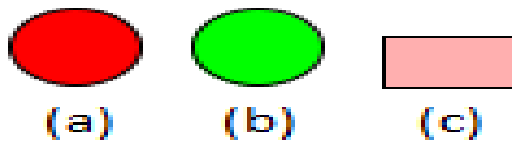
Notasi grafis yang digunakan dalam FamilyFinder menggunakan referensi dari *Graphical For Semantic Web Language* [5] yang telah disesuaikan dengan penggunaan pada aplikasi ini seperti pada Gambar 2. Keterangan Gambar 2 sebagai berikut.

- Instance* untuk setiap anggota keluarga yang mempunyai jenis kelamin pria.
- Instance* untuk setiap anggota keluarga yang mempunyai jenis kelamin wanita.
- Predikat, dimana predikat menyatakan relasi antar anggota keluarga.

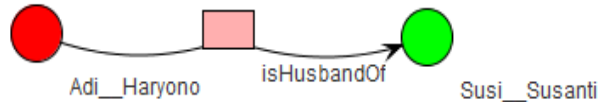
Untuk memodelkan secara grafis, pernyataan “Adi Haryono *is husband of* Susi Susanti” digunakan garis yang merepresentasikan struktur dari pernyataan tersebut. Gambar 3 menunjukkan bahwa Adi Haryono (*instance* dari *Man*) ialah suami Susi Susanti (*instance* dari *Woman*) dimana *isHusbandOf* adalah properti yang menyatakan hubungan relasi antara subjek dan objek.

C. Protege

Protege adalah *open source* ontologi *editor* dan kerangka kerja *knowledge-base* [6]. Protege berguna untuk membuat



Gambar 2. Notasi Grafis



Gambar 3. Contoh Penggunaan Notasi Grafis

struktur kelas, objek properti, data properti, dan lain-lain. Struktur kelas FamilyFinder seperti yang ditunjukkan pada Gambar 4 berisi kelas *Person*. Kelas *Person* memiliki sub kelas *Man* dan *Woman*.

FamilyFinder memiliki objek properti “*isRelationOf*” yang memiliki sub kelas seluruh kosa kata yang dipakai dalam aplikasi FamilyFinder dan memiliki imbuhan “*is...Of*”.

Objek properti FamilyFinder dikelompokkan menjadi lima yaitu *isBloodRelationOf*, *isFosterRelationOf*, *isInLawOf*, *isSpouseOf*, dan *isStepRelationOf*. Tujuan dari pengelompokan tersebut ialah untuk mengetahui relasi *general* anggota keluarga.

D. Jena

Jena menyediakan sekumpulan *tool* dan pustaka Java untuk membantu dalam membuat semantik *web* dan aplikasi yang memiliki data yang terhubung, *tools* dan *server* [7]. Kerangka Jena berisi keterangan sebagai berikut.

1. API untuk membaca, memproses dan menulis RDF data ke dalam *format* XML, N-triples dan Turtle.
2. *Ontology* API untuk memproses OWL dan ontologi RDFS.
3. *Rule-based inference engine* untuk melakukan penalaran RDF dan sumber data OWL.
4. Menyimpan RDF triple dalam jumlah yang besar agar tersimpan secara efisien.
5. SPARQL sebagai alat untuk melakukan *query*.

E. Jung API

Kerangka Kerja Jaringan/Graf *Universal Java* (*Java Universal Network/Graph Framework*, atau yang biasa disebut dengan JUNG) adalah pustaka perangkat lunak yang menyediakan bahasa umum dan dapat dikembangkan untuk pemodelan, analisis, dan visualisasi data yang dapat direpresentasikan dalam graf atau jaringan. Sebagai pustaka sumber terbuka, JUNG menyediakan kerangka kerja untuk analisis dan visualisasi graf [8].

Arsitektur JUNG dirancang untuk mendukung berbagai macam representasi dari entitas dan hubungannya, seperti graf berarah dan tidak berarah, graf *multi-modal*, graf dengan busur paralel, dan *hypergraph*. JUNG juga dapat mengimplementasikan beberapa algoritma umum pada teori graf, seperti fungsi untuk *clustering*, dekomposisi, *optimization*, analisis statistik, dan lain-lain [9].



Gambar 4. Struktur Kelas FamilyFinder

III. METODOLOGI

A. Arsitektur Perangkat Lunak

Aplikasi FamilyFinder dapat digunakan oleh siapa saja yang ingin membuat silsilah keluarganya sendiri. Pengguna juga dapat melakukan pencarian relasi keluarga dan hasil pencariannya ditampilkan dengan menggunakan *query by design*. Berikut ini adalah alur kerja program secara umum seperti yang ditunjukkan pada Gambar 5 sebagai berikut.

- a. *Person* membuat relasi keluarga menggunakan notasi grafis.
- b. *Input* grafis dari *Person* dirubah oleh aplikasi menjadi *query* Jena.
- c. Jena mengeksekusi *query insert* dari *Person* dan menambahkan relasi triplet ke dalam *file* RDF.
- d. *Person* memasukkan visual *query* menggunakan notasi grafis.
- e. Aplikasi membaca *query* visual dari *Person* dan merubahnya menjadi *query* Jena.
- f. Jena mengeksekusi *query select* dari aplikasi dan mengambil data RDF.
- g. Hasil dari *query* Jena diberikan kembali ke aplikasi.
- h. Aplikasi membentuk relasi-relasi keluarga menjadi visual *graph*.

B. Skenario Kasus Penggunaan

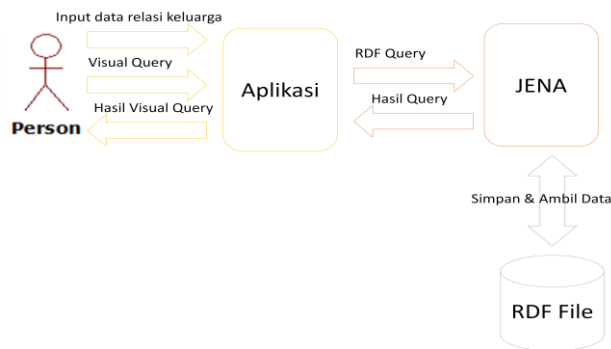
Dalam pendefinisian kebutuhan perangkat lunak, aplikasi ini menggunakan diagram kasus penggunaan. Kasus penggunaan yang digunakan pada sistem perangkat lunak ini seperti yang digambarkan pada Gambar 6. Hal tersebut merupakan kebutuhan fungsional yang berisi kumpulan proses bisnis dalam perangkat lunak yang harus dipenuhi.

Aktor dalam sistem ini adalah *person*. Dimana *person* merupakan seseorang yang melakukan pembuatan atau pencarian silsilah keluarga. *Person* dapat membuat silsilah keluarga, mengubah silsilah keluarga, mencari silsilah keluarga, dan melihat hasil pencarian.

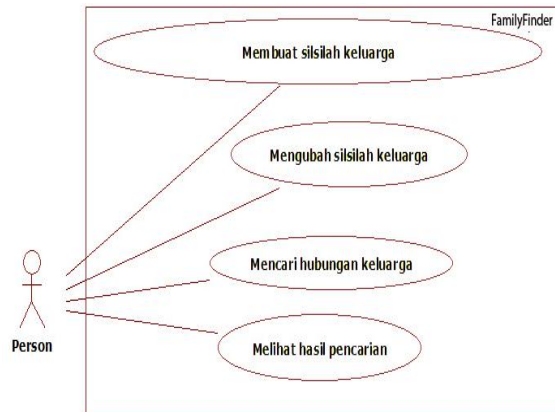
C. Perancangan Kelas Diagram

Perancangan kelas diagram ditentukan dari ringkasan penggabungan dari semua diagram urutan yang telah dibuat. Kelas-kelas yang terlibat pada proses ini adalah kelas *FFMainForm*, *JenaMainSingleton*, *GraphMethod*, *QueryMethod*, *QueryForm*, *PersonForm*, *EditingCheckingPlugin*, *VertexMenuListener*, *MyMouseMenus* dan *PersonForm* seperti pada Gambar 7.

Kelas *FFMainForm*, *QueryForm*, dan *PersonForm* merupakan *presentation layer*. *JenaMainSingleton* merupakan *data layer* yang berhubungan langsung dengan data RDF. *MyProj* dan *MouseMenu* merupakan *layer* yang berisi logika aplikasi. *MouseMenu* dikhususkan untuk logika aplikasi yang



Gambar 5. Alur Kerja Program Secara Umum



Gambar 6. Diagram Kasus Penggunaan Sistem

mengatur visualisasi *graph*. Sedangkan *MyProj* dikhususkan untuk logika pembuatan dan pencarian silsilah keluarga.

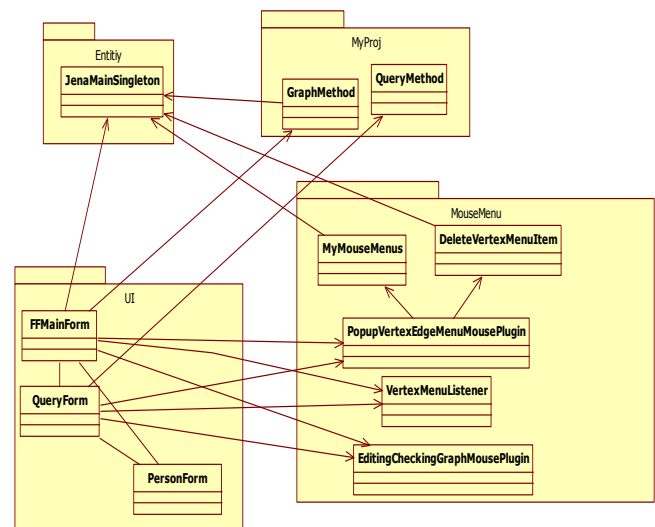
IV. IMPLEMENTASI

Implementasi pertama pada aplikasi ini adalah pembuatan silsilah keluarga. Setelah *person* menambahkan relasi triplet baru, *query* pembuatan relasi triplet akan dijalankan. Sistem akan melakukan pengecekan apakah relasi tersebut valid atau dapat dibuat. Setelah itu sistem akan menambahkan relasi triplet ke dalam *file* penyimpanan relasi silsilah keluarga. Berikut ini adalah kode pembuatan relasi triplet seperti yang ditunjukkan pada Gambar 8.

Implementasi kedua pada aplikasi ini adalah pembuatan *query* pencarian silsilah keluarga. Sistem membuat *query* SPARQL dari notasi grafis pencarian. Setelah didapatkan *query* maka sistem akan mengeksekusi *query* tersebut. Kemudian aplikasi akan mengubah hasil *query* menjadi notasi grafis dan kemudian ditampilkan pada *form* aplikasi. Berikut ini adalah kode pembuatan *query* relasi triplet seperti yang ditunjukkan pada Gambar 9.

V. PENGUJIAN DAN EVALUASI

Pengujian aplikasi silsilah keluarga ini dengan menggunakan sebuah laptop. Pengujian perangkat lunak ini menggunakan metode pengujian *black box* yang berfokus pada kebutuhan fungsional. Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasi dan bekerja seperti yang semestinya.



Gambar 7. Diagram Kelas

```
public Boolean createTriple(String rSor,String prop,String rDest){
    Individual r1=m.getIndividual(NS+rSor);
    Individual r2=m.getIndividual(NS+rDest);
    ObjectProperty objProp = m.getObjectProperty(NS+prop);
    r1.addProperty(objProp, r2);
    saveModel();
}
```

Gambar 8. Fungsi Pembuatan Relasi Triplet

```
public Map<String, List<String>> getResultOfVisualQuery()
{
    String where = getQueryWhereStrNFillQLst();
    String queryJena = JenaMainSingleton.getInstance().queryPrefix +
    getQuerySelectStr() + where;
    //kode lainnya..
    return
    JenaMainSingleton.getInstance().executeQueryInference2Map(queryJena,que
    stions);
}
```

Gambar 9. Fungsi Membuat Dan Mendapatkan Hasil Visual Query

A. Pengujian Membuat Silsilah Keluarga

Pengujian membuat silsilah keluarga memiliki skenario sebagai berikut:

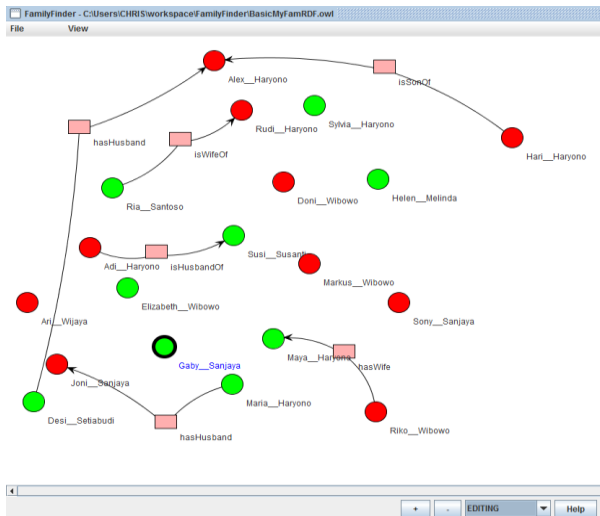
1. program aplikasi dijalankan.
2. memilih menu "File".
3. memilih *submenu* bernama "New Family Tree".
4. memilih memberikan nama *file* pada silsilah keluarga yang akan dibuat.
5. memilih mode "Editing" pada *comboBox*.
6. membuat anggota keluarga baru dengan menambahkan instansiasi *Man* atau *Woman*.
7. *drag mouse* dari instansiasi subjek lalu *drop* pada instansiasi objek.
8. memberikan nama properti pada *dialogBox* yang muncul.
9. memilih tombol "OK".

Pengujian ini berhasil karena relasi silsilah keluarga baru dapat terbuat seperti yang ditunjukkan pada Gambar 10.

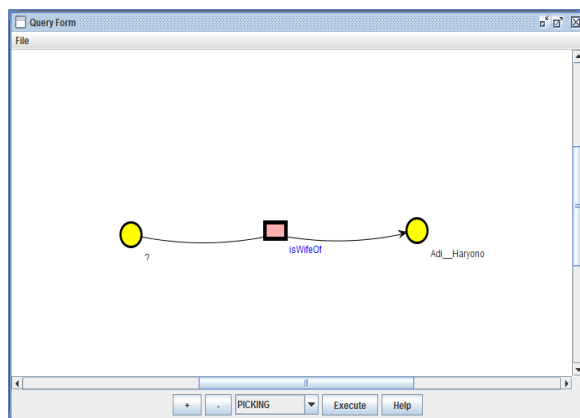
B. Pengujian Query Pencarian Silsilah Keluarga

Pengujian *query* pencarian silsilah keluarga memiliki skenario sebagai berikut:

1. memilih menu "View".
2. memilih salah satu *submenu template query*.



Gambar 10. Tampilan Hasil Pembuatan Silsilah Keluarga



Gambar 11. Tampilan Form Query Pencarian

3. klik kanan pada *node* yang telah ada untuk mengganti subjek, predikat, dan objek sesuai kebutuhan *person*.
4. tekan tombol "OK" pada *dialogBox* yang muncul.
5. tekan tombol "Execute".

Pengujian ini berhasil karena sistem dapat menampilkan *form query* pencarian seperti yang ditunjukkan pada Gambar 11. Kemudian, sistem dapat menampilkan hasil pencarian keluarga dengan benar seperti yang ditunjukkan pada Gambar 12.

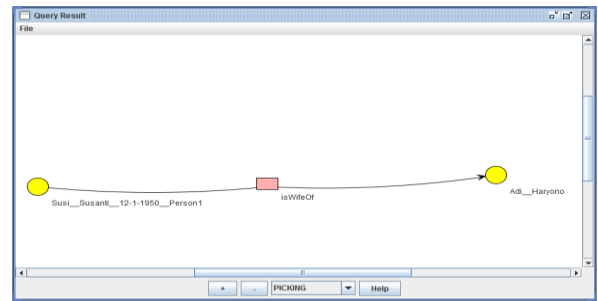
C. Analisis Hasil Pengujian

Berdasarkan hasil pengujian fungsionalitas perangkat lunak FamilyFinder, semua skenario berhasil dilakukan dan fitur tersebut telah teruji dapat membuat dan mencari silsilah keluarga dengan relasi keluarga yang khas Indonesia.

VI. KESIMPULAN

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba yang dilakukan, kesimpulan yang dapat diambil berdasarkan rumusan masalah yang ada adalah sebagai berikut ini.

1. Aplikasi ini menggunakan masukan *query by design* untuk membuat relasi keluarga.



Gambar 12. Tampilan Hasil Pencarian Silsilah Keluarga

2. Aplikasi FamilyFinder dapat menampilkan hasil *query person* dengan benar.
3. *Person* dapat melihat hasil *query* relasi keluarga melalui notasi grafis.

UCAPAN TERIMA KASIH

Penulis C.C. mengucapkan terima kasih kepada Tuhan Yang Maha Esa, kedua orangtua dan keluarga penulis, dosen pembimbing, seluruh dosen Teknik Informatika ITS, kerabat dekat, serta berbagai pihak yang telah membantu penulis dalam menyelesaikan penelitian ini.

DAFTAR PUSTAKA

- [1] "Drawing a Family Tree," [Online]. Available: <http://njaes.rutgers.edu/healthfinance/pdfs/family-medical-tree.pdf>. [Diakses pada tanggal 6 Juli 2013].
- [2] M. C. Daconta, The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management.
- [3] G. A. a. F. v. Harmelen, Semantic Web Primer, London: MIT Press.
- [4] N. Ibrahim, Pengembangan Aplikasi Semantic Web Untuk Membangun Web yang Lebih Cerdas, 2007.
- [5] D. Siahaan, Graphical Notations For Semantic Web Language, Surabaya, 2006.
- [6] "Introduction to Ontologies with Protege," [Online]. Available: <https://wiki.csc.calpoly.edu/OntologyTutorial/wiki/IntroductionToOntologiesWithProtege>. [Diakses pada tanggal 4 Juli 2013].
- [7] Apache Jena, "Jena Ontology API," [Online]. Available: <http://jena.apache.org/documentation/ontology/>. [Diakses pada tanggal 4 Juli 2013].
- [8] "tentang pustaka JUNG," [Online]. Available: <http://jung.sourceforge.net/>. [Diakses pada tanggal 5 Juli 2013].
- [9] "Dokumentasi Jung," [Online]. Available: <http://jung.sourceforge.net/doc/api/index.html>. [Diakses pada tanggal 7 Juli 2013].