Journal of Data Analysis
Vol.2, No.1, June 2019, p. 01-16

ONTODB : Aplikasi untuk Transformasi Ontologi OWL ke Basis Data Relasi SQL

Amalia Mabrina Masbar Rus[1 *], Zulaiha Ali Othman[2]

[1] Department of Informatics, Universitas Syiah Kuala, Indonesia
[2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia
E-mail: amaliammr@unsyiah.ac.id*

Abstrak

OntoDB Application Tool adalah aplikasi desktop yang dikembangkan dengan menggunakan bahasa pemrograman Java. Aplikasi ini dikembangkan sebagai alat untuk mentransformasikan file ontologi dalam format OWL ke file kode basis data SQL. Transformasi OWL ke SQL diperlukan untuk menyimpan ontologi OWL ke dalam basis data, sehingga memudahkan dalam pencarian data dan digunakan sebagai sumber data untuk aplikasi. Namun, file ontologi yang berisi sejumlah besar kelas, properti, dan *instance* membuat pengembangan aplikasi sulit dilakukan apabila hanya menggunakan beberapa komponen ontologi. Oleh karena itu, aplikasi OntoDB dikembangkan untuk memenuhi kebutuhan akan sebuah aplikasi yang dapat menyimpan sebagian ontologi ke dalam basis data. Aplikasi ini menampilkan isi komponen ontologi utama seperti kelas, properti dan *instance* sehingga pengguna dapat dengan mudah memilih komponen mana yang akan ditransformasikan. Selanjutnya, aplikasi ini akan menghasilkan file SQL berdasarkan isi ontologi yang dipilih. Kemudian, file SQL yang dihasilkan dapat diimpor ke dalam sistem manajemen basis data, seperti MySQL, untuk dijadikan sumber data bagi sebuah aplikasi.

Informasi Artikel

Sejarah Artikel:
Diajukan Feb 19, 2019
Diterima Jun 29, 2019

Kata Kunci:
Transformasi Ontologi
Penyimpanan Ontologi
OWL
SQL
Web Semantik

Abstract

OntoDB Application Tool is a desktop application developed using Java programming language. This application was developed as a tool to transform the ontology file in OWL format to an SQL basis data code file. Transforming OWL to SQL is necessary in order to store the OWL ontology into the basis data, thus makes it easier to *query* and to be used as a source of data for an application. However, the ontology file containing a large number of classes, properti, and *instance* make it difficult for the developers to develop an application which only using some components of the ontology. Therefore, OntoDB application was developed to meet the need for a tool that can store a part of ontology into the basis data. This application will display the contents of the main ontology components such as classes, properti and *instance* so that the user can easily select which components that will be transformed. Further, this application will generate an SQL file based on the selected content of ontology. Then, the generated SQL file can be imported into a basis data management system, such as MySQL, to be used as the source of data for an application.

Keyword:
Ontology transformation
Storing ontology
OWL
SQL
Semantic web

1. Pendahuluan

World Wide Web Consortium (W3C) telah mengusulkan penambahan kosakata metadata yang dapat membuat situs web menjadi situs data dengan menggunakan kosa kata RDF, RDF, dan OWL [1]. Namun, karena file kosakata ini berbentuk XML, pengolahan data menggunakan format ini cukup sulit. Hal ini dikarenakan file dalam bentuk flat file tidak memberikan kemudahan dari segi skalabilitas, distribusi, dan mencari data (*querying data*) [2].

Untuk mengatasi hal ini, beberapa penelitian seperti [2]–[6] menyarankan untuk menyimpan data ontologi online OWL dalam basis data. Basis data memiliki rekor yang bagus untuk menyimpan data dalam jumlah besar dan memfasilitasi *query* pada data. Basis data menyediakan teknologi, daya tahan dan kehandalan dalam jangka panjang [2]. Namun, saran aplikasi untuk pemetaan ontologi ke basis data tidak memberikan kemudahan untuk memetakan sebagian saja isi ontologi ke basis data.

Ontologi terdiri dari puluhan kelas dan puluhan atau bahkan ratusan *instance*. Hal ini dibuat agar secara umum ontologi bisa diterapkan dalam konteks apapun. Namun, untuk pengembangan aplikasi yang hanya ingin menyelesaikan konteks tertentu dan ingin menggunakan ontologi sebagai sumber datanya, penggunaan ontologi umum ini tidak efektif.

Hal ini dikarenakan beberapa konteks lain yang terlalu umum dan tidak terkait langsung dengan aplikasi ini juga harus dipetakan ke basis data. Dengan demikian, pengguna aplikasi tidak dapat memilih kelas, properti atau *instance* yang dibutuhkannya saja. Oleh karena itu aplikasi baru yang dapat memetakan sebagian ontologi ini ke basis data untuk digunakan dalam pengembangan aplikasi tertentu sangat dibutuhkan.

Penyimpanan Ontologi berupa file flat menyebabkan ontologi sulit untuk di-*query* dan dibaca oleh pengguna. Dalam bentuk file flat, ontologi ditampilkan dalam bentuk kode XML yang panjang. Sulit bagi pengguna file ini untuk melihat isi ontologi ini secara keseluruhan. Pengguna sulit mengetahui data ontologi yang disimpan seperti kelas, properti dan contoh serta pengaturan dan hubungan di antara keduanya. Oleh karena itu diperlukan alat yang bisa menampilkan isi dan kaitan antara komponen ontologi seperti kelas, properti dan *instance*.

Selain itu, penyimpanan ontologi berupa file flat juga tidak memberikan kemampuan untuk mendapatkan informasi dengan cepat melalui *query*. Untuk itu, ontologi harus disimpan di tempat penyimpanan yang memungkinkan *query* pada ontologi dengan menyimpannya di basis data.

Ketika melakukan pengembangan basis data untuk aplikasi tertentu juga muncul masalah dalam mendesain tabel dan field yang akan digunakan untuk menyimpan data. Pertukaran data antara satu basis data dan lainnya sulit karena adanya penamaan tabel dan field yang berbeda walaupun isi data yang tersimpan mengacu pada hal yang sama. Misalnya, di basis data 1 terdapat tabel 'Siswa' dengan field 'Student ID'. Sedangkan di basis data 2 terdapat tabel 'Student' dengan field 'IDSiswa'. Perbedaan penamaan ini membuat perlunya program tambahan untuk menyamakan dua konvensi penamaan untuk pertukaran data.

Oleh karena itu diperlukan penamaan standar dalam merancang basis data. Ontologi dapat digunakan sebagai solusi untuk menstandarkan penamaan tabel dan field dalam basis data. Dengan menggunakan ontologi yang sama untuk mengembangkan basis data, basis data yang dihasilkan akan memiliki penamaan yang sama dan dapat memudahkan pertukaran data. Oleh karena itu, alat untuk memetakan ontologi ke basis data harus dikembangkan untuk memudahkan penyimpanan data dalam basis data.

Namun, aplikasi yang ada saat ini tidak memungkinkan untuk memetakan hanya sebagian isi ontologi. Sebagai contoh, pembangun basis data hanya ingin mengembangkan basis data

Factory Worker. Namun, ontologi yang ada, misalnya *Work Ontology*, tidak hanya menyimpan informasi tentang *Factory Worker*, tapi juga informasi tentang pekerjaan lain. Oleh karena itu, diperlukan aplikasi OntoDB ini untuk dapat memetakan sebagian saja isi ontologi menjadi SQL. Aplikasi ini dapat menampilkan isi ontologi dengan menggunakan antarmuka, sehingga dapat memudahkan pengguna dalam memilih kelas, properti, dan *instance* tertentu saja dari ontologi yang ingin ditransformasi.

2. Tinjauan Pustaka

Setelah menganalisis beberapa alat untuk mentransformasi ontologi kepada basis data atau menyimpan ontologi ke dalam basis data, terdapat 2 teknik yang digunakan. Teknik tersebut ialah teknik mentransformasi dan teknik menyimpan dalam basis data yang sudah ditetapkan.

2.1 Teknik Mentransformasi

Teknik pertama yaitu teknik mentransformasikan ontologi. Dalam teknik ini, setiap kelas dipetakan menjadi tabel, properti menjadi *field* dan *instance* menjadi rekod dalam tabel. Hubungan antar-kelas akan dipetakan menggunakan konsep *Foreign Key*. Hubungan antara kelas dengan *instance* akan dipetakan menggunakan konsep tabel. Suatu *instance* dari sebuah kelas akan menjadi rekod dalam tabel kelas tersebut. Data dari *instance* tersebut akan disimpan sesuai dengan properti-properti bagi kelas tersebut. Contoh perangkat lunak yang menggunakan teknik ini ialah QUALEG DB [2] dan [7].

2.2 Teknik Menyimpan dalam Basis Data

Teknik yang kedua adalah dengan mengumpulkan semua kelas dalam satu tabel, properti dalam tabel yang lain, dan *instance* dalam tabel yang lain pula. Untuk menghubungkan antara satu kelas dengan properti dan *instance*, digunakan satu tabel penghubung. Setiap kelas, properti dan *instance* memiliki ID tersendiri. Dengan demikian, dalam tabel penghubung, yang disimpan adalah ID dari kedua benda yang memiliki hubungan saja. Dengan menggunakan cara ini, tabel dalam basis data telah ditetapkan terlebih dahulu dan tidak akan berubah atau bertambah, hanya datanya saja yang akan berubah. Contoh aplikasi yang menggunakan teknik ini ialah OntRel [3] dan OWL2DB [4].

Setelah menganalisis kedua teknik ini, diketahui bahwa teknik menyimpan dalam basis data adalah lebih efisien dan lebih baik daripada teknik trnasformasi. [3] telah melakukan pengujian terhadap aplikasi OntRel dengan menggunakan teknik pengujian LUBM (Leigh Univeristy Benchmark) dan UOBM (University Ontology Benchmark). Dari hasil pengujian ini diketahui bahwa aplikasi ini lebih baik dari segi waktu pemuatan (*loading time*) data ontologi ke basis data dan kelengkapan dalam menjawab pertanyaan (*query*) dalam basis data.

Pembangunan dan pengembangan aplikasi OntoDB menggunakan aturan-aturan ini karena teknik yang digunakan pada aplikasi OntRel lebih baik dibandingkan teknik yang lain. Teknik pengujian menggunakan data LUBM juga diterapkan pada aplikasi OntoDB untuk menguji kemampuan aplikasi ini dalam menyimpan ontologi ke dalam basis data. Berikut adalah tabel perbandingan fitur aplikasi QULLEG DB, OntRel, dan OntobDB.

Tabel 1 Perbandingan fitur aplikasi QUALLEG DB, OntRel, dan OntoDB

Fitur	QUALLEG DB	OntRel	OntoDB
Kesashihan (*Validity*)	Tidak	Ya	Ya
Skalabilitas (*Scalability*)	Tidak	Ya	Ya
Kelengkapan	Tidak	Ya	Ya
Konsistensi	Tidak	Ya	Ya
Menampilkan Ontologi	Tidak	Tidak	Ya
Dapat mentransformasi sebagian isi ontologi	Tidak	Tidak	Ya
Teknik Transformasi	Teknik 1	Teknik 2	Teknik 2

2.3. Aturan-aturan Transformasi Ontologi ke Basis Data

Aturan-aturan transformasi ontologi ke basis data yang digunakan dalam aplikasi alat transformasi ontologi ke basis data (OntoDB) berikut berdasarkan Teknik Menyimpan dalam Basis Data yang digunakan pada aplikasi OntRel [3]. Terdapat 16 Aturan yang diterapkan dalam mentansformasi ontologi ke dalam basis data. Aturan-aturan tersebut dijelaskan sebagai berikut.

Aturan 1: Aturan untuk Ontologi Baru dan Induk Ontologi

Untuk setiap ontologi baru, entri ontologi akan disimpan di dalam tabel Onto_tbl. Tabel ini berisi ID ontologi, domain, URI, keterangan, versi, label, komen, dan versi lama.

Aturan 2: Aturan untuk Konsep Baru

Untuk setiap konsep baru, diberikan ID unik dan entri konsep ini disimpan dalam tabel Concept_tbl dengan ID Ontologi sebagai kunci asing bagi setiap konsep. Concept_tbl juga berisi nama konsep, URI, label dan komen.

Aturan 3: (Aturan bagi Properti dan Karakteristik Properti)

Entry bagi setiap properti yang digunakan pada konsep tertentu dimasukkan ke dalam tabel Properti_tbl. Properti-properti tersebut dapat berupa *Datatype* Properti atau Objek Properti. Sebuah ID unik diberikan kepada masing-masing properti dan bertindak sebagai Kunci Utama (*Primary Key*). Tabel Properti_tbl berisi nama properti, jenis properti, *Domain*, *Range* dan *Range value*. Jenis properti menunjukkan properti tersebut adalah Datatype Properti atau Objek Properti. Jika ada batasan (*constraint*) seperti properti fungsional, properti invers fungsional, properti simetri atau properti transitif yang berlaku terhadap properti itu, ID properti akan dimasukkan ke dalam tabel propertyCharacteristics_tbl. Dalam tabel ini masing-masing properti ditandakan dengan sebuah bendera unik untuk mewakili jenis batasannya.

Aturan 4: (Aturan bagi Objek Properti)

Dalam tabel Properti_tbl, jika jenis properti menyatakan bahwa properti tersebut adalah Objek Properti, maka *Domain* dari properti ini akan berisi ID Konsep. Begitu juga halnya dengan *Range*, akan berisi ID Konsep yang merupakan kisaran nilai bagi properti ini.

Aturan 5: (Aturan Properti Datatype)

Jika jenis properti merupakan jenis Properti Datatype, maka *Domain* akan tetap diisi dengan ID Konsep sedangkan *Range* akan diisi dengan ID dari *XSD Datatype*.

Aturan 6: (Aturan bagi {SubPropertyOf, InverseProperty, EquivalentProperty})

Jika suatu properti merupakan SubPropertyOf, InverseProperty, atau EquivalentProperty dari beberapa properti lain, maka entri hubungan tersebut akan dimasukkan pada tabel SubPropertyOfAxiom_tbl, InverseProperty_tbl atau EquivalentProperty_tbl sesuai dengan jenis hubungannya.

Aturan 7: (Aturan bagi {SubClassOf, ComplementOf, EquivalentOf, DisjointWith} yang diketahui)

Jika ada hubungan SubClassOf, ComplementOf, EquivalentOf, atau DisjointWith antara dua konsep, maka untuk menjaga hubungan tersebut, entri hubungan akan dimasukkan ke dalam tabel SubClassOfAxiom_tbl, ComplementOfAxiom_tbl, EquivalentOfAxiom_tbl atau DisjointWithAxiom_tbl sesuai dengan jenis hubungannya dimana semua ID konsep bertindak sebagai Kunci Asing (*foreign key*).

Aturan 8: (Aturan bagi {SubClassOf, ComplementOf, EquivalentOf, DisjointWith} yang tidak diketahui)

Jika kelas adalah SubClassOf, ComplementOf, EquivalentOf atau DisjointWith dengan kelas Anonim, maka untuk menjaga hubungan entri dari Anonim Induk Konsep ID, Pelengkap Anonim Konsep ID, ID Konsep Anonim Setara atau Konsep ID tanpa nama menguraikan dilakukan terhadap ID Konsep di SubClassOfAxiom_tbl, ComplementOfAxiom_tbl, EquivalentOfAxiom_tbl atau DisjointWithAxiom_tbl masing-masing dimana semua ID konsep bertindak sebagai Kunci Asing.

Aturan 9: (Aturan Irisan dengan Konsep Anonim)

Jika terdapat hubungan irisan antara Konsep Anonim, untuk menyimpan hubungan ini, entri dari beberapa ID Konsep Anonim akan disimpan dalam ID Konsep di Intersection_tbl, dimana ID Konsep Anonim adalah beberapa konsep Anonim yang mendefinisikan hubungan irisan. Dalam kasus ini, IsAnonymous field akan menjadi benar dalam tabel Intersection_tbl.

Aturan 10: (Aturan bagi Konsep Anonim dan Enumeration Anonim)

Untuk setiap konsep, ID unik akan diberikan kepada setiap konsep Anonim baru dan disimpan dalam tabel AnonymousConcept_tbl dengan ID Ontologi sebagai kunci asing bagi masing-masing konsep. Konsep Anonim dapat berupa Anonim {Union, Intersection, Complement, Enumeration atau Restriction}. Oleh sebab itu, AnonymousConcept_tbl juga berisi ID dan Bendera bagi setiap jenis hubungan anonim tersebut. Jika bendera bernilai benar, maka ID akan

berisi ID yang akan menjadi kunci asing dari Anonymouse{Union, Intersection, Complement, Enumeration atau Restriction}_tbl. Jika Konsep Anonim adalah Enumeration dari individu-individu lain, maka ID unik akan diberikan bagi setiap Individu dan entri tentang hubungan ini akan disimpan dalam tabel AnonymousEnumeration_tbl dimana AenumID bertindak sebagai kunci asing.

Aturan 11: (Aturan bagi Konsep Anonim{Union, Intersection, Complement})

Jika terdapat konsep anonim yang terbentuk disebabkan adanya hubungan Union, Intersection, atau Complement, maka setiap AUnionID, AIntersectionID, atau AComplementID akan menyimpan ID dari konsep Anonim atau konsep yang telah diketahui. IsAnonymous field menunjukkan apakah konsep tersebut konsep anonim atau bukan. Semua hubungan ini akan disimpan dalam tabel AnonymousUnion_tbl, AnonymousIntersection_tbl, atau AnonymousComplement_tbl.

Aturan 12: (Aturan bagi Batasan Anonim)

Untuk semua batasan dalam dokumen OWL yang berlaku pada konsep yang diketahui atau konsep anonim, entri batasan akan disimpan dalam tabel AnonymousRestriction_tbl. Tabel ini berisi ArestrictionID unik, Jenis Batasan, ID Properti yang dibatasi, ID Objek Konsep dan nilai batasan.

Aturan 13: (Aturan bagi Individu dan Individu Anonim)

Untuk semua individu yang ada dalam ontologi, suatu ID yang unik akan diberikan bagi masing-masing individu dan disimpan dalam tabel Individual_tbl. Kombinasi ID Konsep dan ID Individu menyatakan bahwa suatu individu adalah kepunyaan dari suatu konsep tertentu. Jika individu adalah individu anonim, maka entrinya akan disimpan dalam tabel AnonimIndividual_tbl yang menyimpan ID Individu dan ID Konsep ID untuk mewakili hubungan ini. Jika field IsAnonymous bernilai benar, maka individu terebut adalah individu anonim.

Aturan 14: (Aturan Individu{sameAs, differentFrom, allDifferent})

Individu dapat memiliki hubungan sameAs, differentFrom dan allDifferent dengan individu lainnya. Jika ada hubungan sameAs antara dua individu, maka kedua ID Individu ini akan disimpan dalam tabel IndividualSameAs_tbl. Untuk hubungan DifferentFrom dan AllDifferent antara dua individu akan disimpan dalam tabel DifferentFrom_tbl, di mana kedua ID Individu bertindak sebagai kunci asing dari Individual_tbl.

Aturan 15: (Aturan Properti Individu dan Hubungan Individu)

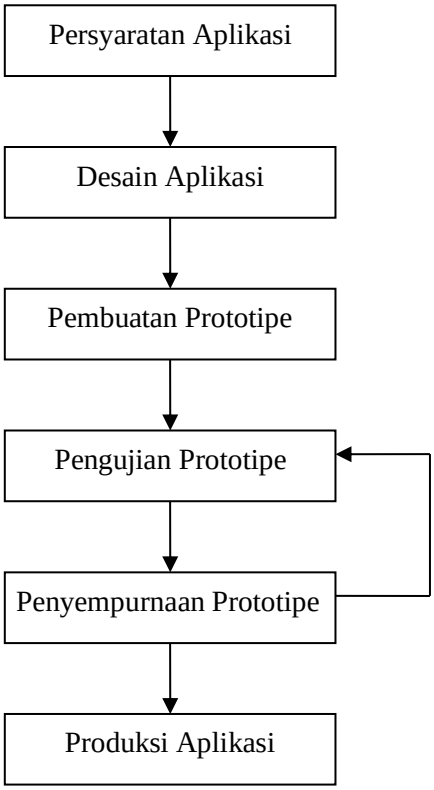
Dua individu boleh mempunyai hubungan properti antara satu sama lain. Hubungan ini akan disimpan dalam tabel IndividualProperty_tbl dengan field IndividualID, PropertyID dan RangeIndividualID dimana PropertyID menunjukkan hubungan antara IndividualID dan RangeIndividualID.

Aturan 16: (Aturan Anotasi Properti)

Anotasi disimpan dalam tabel annotations_tbl dimana ID Anotasi adalah kunci utamanya. IdentificationFlag dapat mewakili Konsep, Properti atau Individu. AnnotationProp dan Nilai Anotasi mewakili penjelasan terhadap ConstructID yaitu kunci asing yang diidentifikasi oleh field identificationID.

3. Metode Penelitian

Metode penelitian yang digunakan dalam pembangunan aplikasi OntoDB ini adalah Model Prototipe. Gambar 1 menunjukkan alur metode penelitian yang digunakan dalam pembangunan aplikasi OntoDB ini.



Gambar 1 Model Prototipe (Sumber: [8])

Terdapat beberapa tahapan dalam pembangunan aplikasi OntoDB dengan penjelasan sebagai berikut.

1. Persyaratan Aplikasi
   Pada tahap ini dilakukan kajian pustaka untuk menentukan teknik yang terbaik yang akan diaplikasikan pada aplikasi OntoDB. Pada tahap ini pula dilakukan pengumpulan informasi untuk menentukan persyaratan-persyaratan yang diperlukan untuk membangun aplikasi OntoDB ini.
2. Desain Aplikasi
   Pada tahap ini, dilakukan pembuatan desain aplikasi OntoDB dalam bentuk Use Case Scenario, Class Diagram, Entity Relationship Diagram, dan Data Dictionary.

3. Pembuatan Prototipe
   Pada tahap ini prototipe aplikasi dibangun menggunakan IDE Eclipse dan bahasa pemograman JAVA serta menggunakan beberapa library Java yang terkait dengan ontologi seperti Jena Library dan plug-in terkait pembangunan aplikasi yaitu Rich Client Platform. Aplikasi Protégé dan TopBraid Composer juga digunakan untuk menyunting file ontologi dan sebagai rujukan dalam mendesain cara aplikasi memaparkan isi ontologi.
4. Pengujian Prototipe
   Pengujian dilakukan dengan menggunakan 3 ontologi. Ketiga isi ontologi ini akan ditransformasikan seluruhnya ke dalam kode basis data SQL. Selanjutnya dilakukan pengujian terhadap salah satu ontologi untuk menguji kemampuan aplikasi dalam mentransformasikan sebagian saja isi dari ontologi.
5. Penyempurnaan Prototipe
   Pada tahap ini dilakukan penyempurnaan prototipe apabila dari hasil pengujian didapatkan bahwa aplikasi masih belum dapat mencapai target yang telah ditetapkan
6. Produksi Aplikasi
   Pada tahap ini aplikasi sudah dapat mencapai target yang telah ditetapkan dan dapat dijalankan dengan baik.

Tabel 2 menunjukkan informasi tentang ontologi yang digunakan untuk menguji aplikasi OntoDB. Terdapat 3 ontologi yang digunakan yaitu University Ontology, Pizza Ontology, dan Kennedys Ontology. University Ontology merupakan ontologi yang diperoleh dari sistem pengujian LUBM (Leigh University Benchmark). Ontologi ini telah digunakan secara umum untuk menguji aplikasi-aplikasi penyimpanan ontologi dengan jumlah *instance* yang banyak.

Pizza Ontology merupakan ontologi yang memiliki banyak hubungan axiom dan hubungan antar kelas tanpa nama. Ontologi ini dipilih untuk menguji kemampuan aplikasi OntoDB dalam mentransformasi bentuk hubungan axiom dan hubungan dengan kelas tanpa nama.

Kennedys Ontology merupakan suatu ontologi sederhana yang menyimpan informasi hubungan keluarga Kennedy. Ontologi ini dipilih untuk menguji kemampuan aplikasi OntoDB dalam mentransformasi ontologi dengan jumlah *instance* yang kecil (di bawah 200 *instance*).

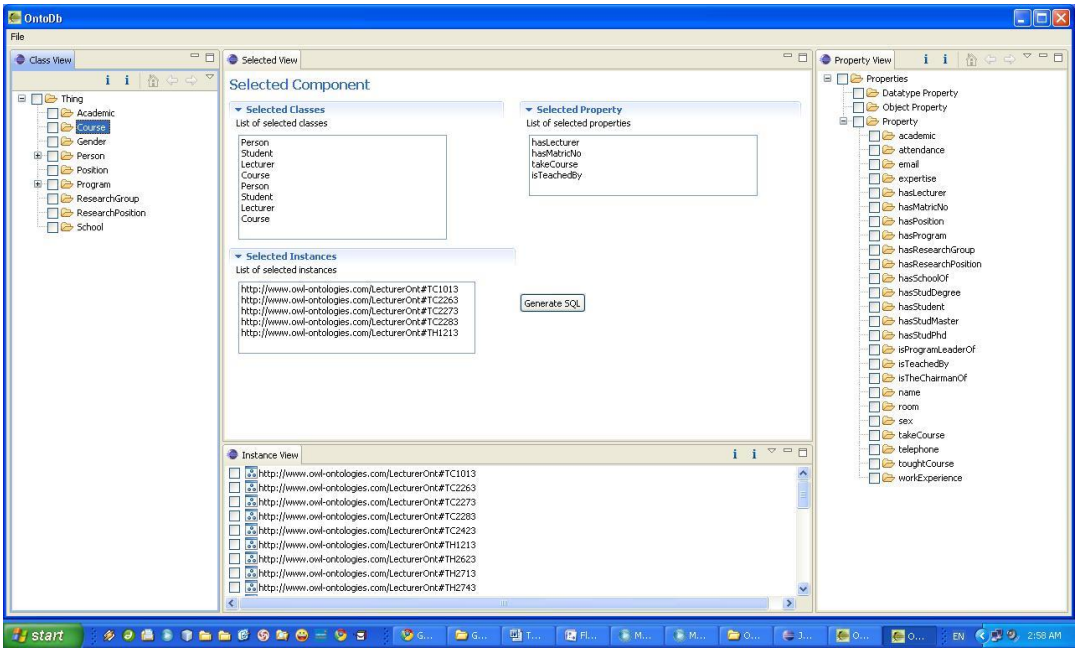
Tabel 2. Perbandingan isi ontologi yang digunakan untuk pengujian aplikasi OntoDB

Hal	University Ontology	Pizza Ontology	Kennedys Ontology
Nama File	University0.0.owl	Pizza.owl	kennedys.owl
Sumber Ontologi			
Deskripsi	Ontologi ini berisi informasi tentang sebuah universitas. Dalam ontologi ini terdapat kelas-kelas seperti orang, jabatan, publikasi, dan lain-lain	Ontologi ini berisi informasi tentang pizza seperti jenis-jenis pizza berdasarkan toppingnya dan lain-lain	Ontologi ini berisi informasi tentang keluarga Kennedy seperti tanggal lahir, tanggal meninggal, hubungan keluarga seperti ayah, ibu, anak, istri atau suami dari anggota keluarga Kennedy

Hal	University Ontology	Pizza Ontology	Kennedys Ontology
Jumlah baris	11341	6858	1638
Jumlah Kelas	43	99	9
Jumlah Object Properti	30	8	16
Jumlah Datatype Properti	7	0	8
Jumlah *Instance*	1657	5	135
Jumlah Kelas Tanpa Nama	14	232	2
Jumlah Kelas Intersection Tanpa Nama	6	15	1
Jumlah Enumeration Tanpa Nama	0	1	0
Jumlah Kelas Complement Tanpa Nama	0	3	0
Jumlah Kelas Restriction	8	188	0
Jumlah Hubungan Equivalent Class	0	15	2
Jumlah Hubungan Intersection	0	0	0
Jumlah Hubungan Disjoint With	0	796	0
Jumlah Hubungan Enumeration	0	1	0
Jumlah Hubungan SubClassOf	38	259	10
Jumlah Hubungan SubPropertyOf	5	4	8
Jumlah Hubungan InverseOf	3	6	1
Jumlah Hubungan Equivalent Properti	0	0	1
Jumlah Hubungan Different From	0	0	0
Jumlah Hubungan Same As	0	0	2
Jumlah Hubungan Imported Ontology	1	0	0

4. Hasil dan Pembahasan

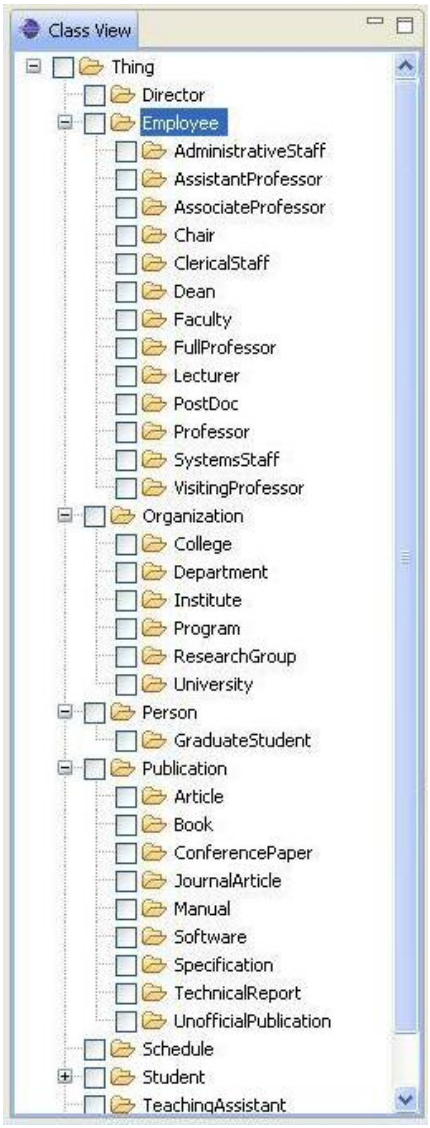
Gambar 2 menunjukkan hasil aplikasi OntoDB yang telah dibangun. Dari gambar berikut terlihat bahwa pengguna dapat melihat isi dari ontologi dengan jelas. Di sebelah kiri, aplikasi memaparkan daftar kelas yang ada dalam ontologi tersebut. Di sebelah kanan terdapat daftar properti yang dapat dipilih, sedangkan di tengah bawah dipaparkan daftar *instance* yang ada dalam ontologi tersebut. Di bagian tengah atas terdapat area tempat memilih kelas, properti dan *instance* yang akan ditransformasikan ke dalam bentuk kode SQL.



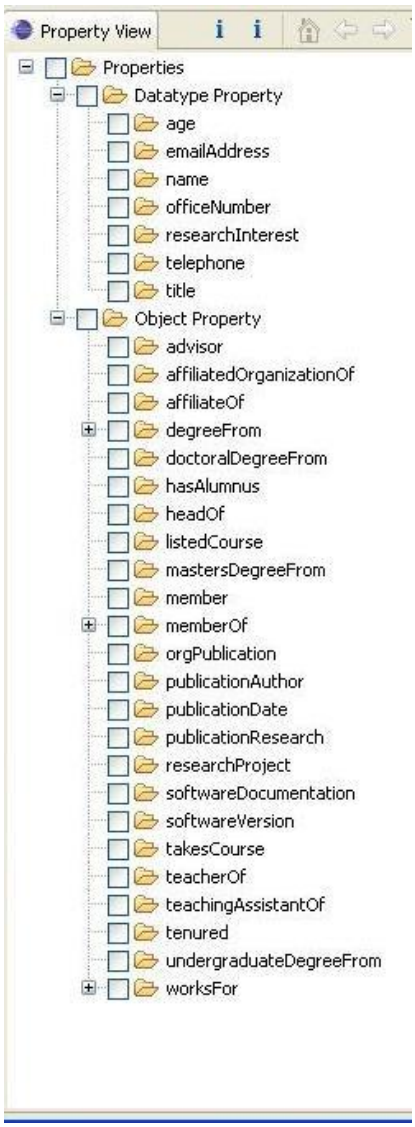
Gambar 2 Antarmuka Utama Aplikasi OntoDB

Gambar 3 dan 4 menunjukkan tampilan Tab Kelas, Properti dan *Instance* dengan lebih jelas. Dapat dilihat bahwa pada Tab Kelas, hubungan antara kelas dan subkelas ditampilkan dalam bentuk struktur pohon (*tree structure)*. Hal ini memudahkan pengguna dalam melihat isi kelas yang ada di dalam file *flat* ontologi dan hubungannya dengan kelas yang lain. Begitu pula dengan Tab Properti yang menampilkan properti dan subproperti dalam bentuk struktur pohon.

Kotak centang yang terdapat di samping nama setiap kelas, properti atau *instance* digunakan untuk memilih kelas, properti atau *instance* tersebut. Komponen-komponen yang dipilih akan ditampilkan pada area Tab Selected (Gambar 5). Untuk menghasilkan kode basis data SQL, pengguna dapat menekan tombol "Generate SQL" dan aplikasi akan memaparkan *window* memilih lokasi file SQL seperti yang ditampilkan oleh Gambar 6.

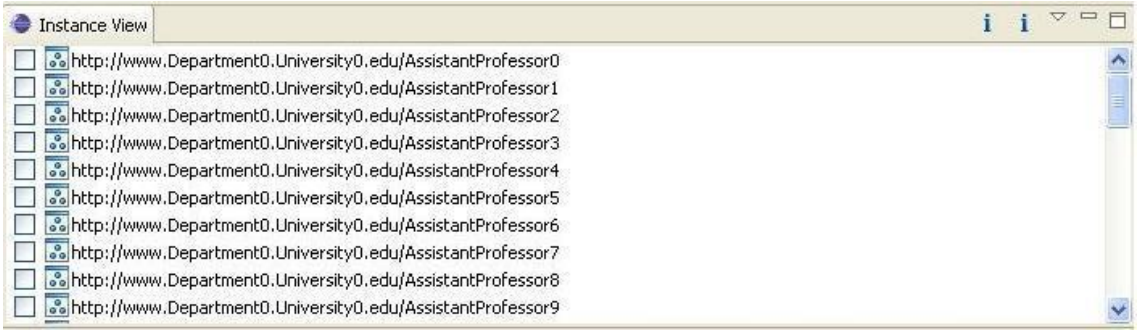


(a)

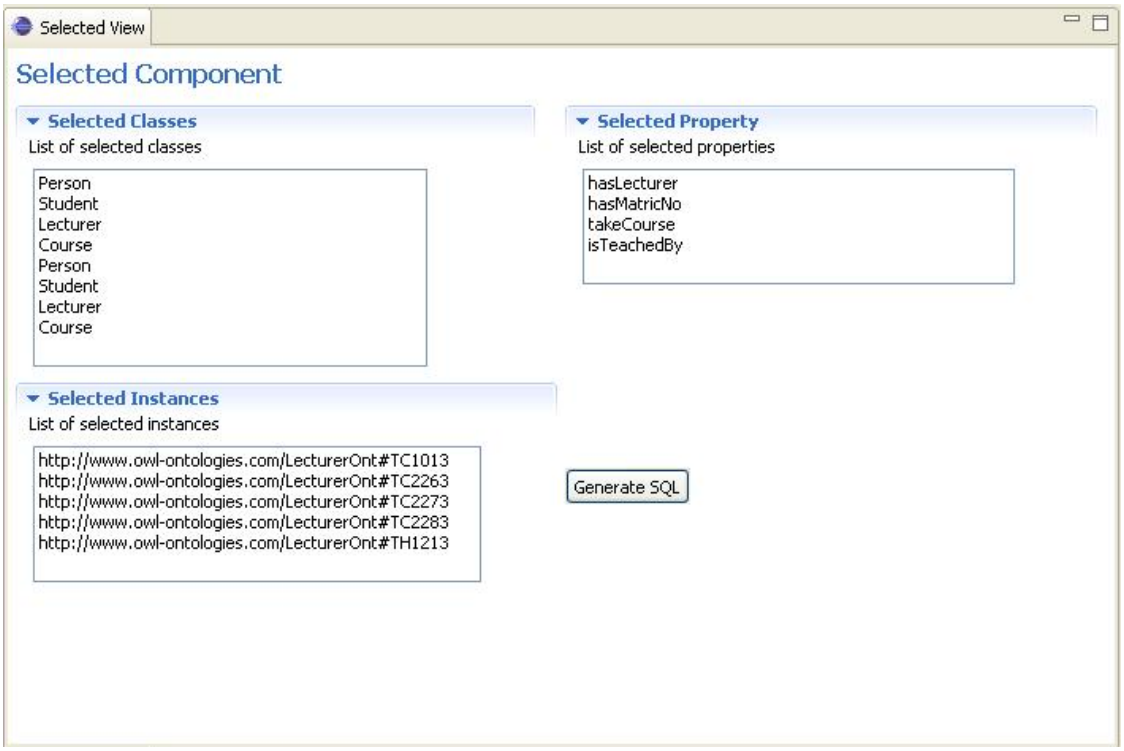


(b)

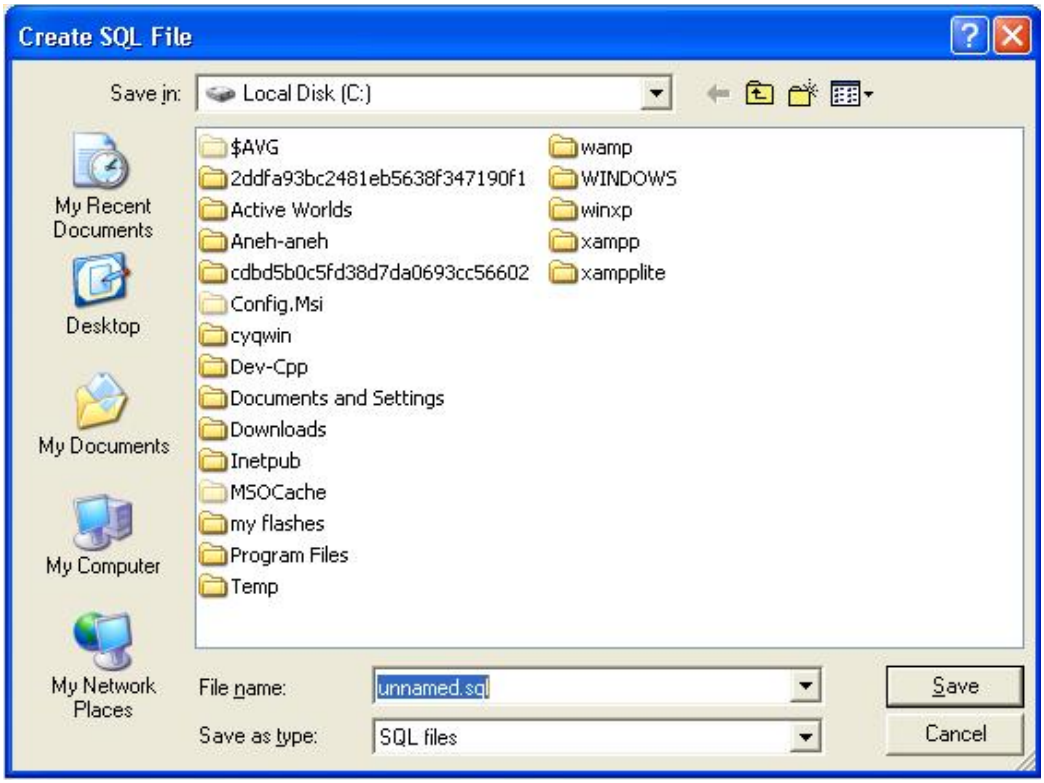
Gambar 3 Antarmuka Tab Kelas (a) dan Tab Properti (b)



Gambar 4 Antarmuka Tab *Instance*



Gambar 5 Antarmuka Tab Selected



Gambar 6 Antarmuka lokasi file SQL

4.1 Hasil Pengujian Mentransformasi Seluruh Isi Ontologi

Pengujian ini dijalankan dengan menggunakan tiga ontologi yaitu University Ontologi, Pizza Ontology, dan Kennedys Ontlogy. Pada pengujian ini seluruh isi ontologi ditrasnformasikan ke dalam kode basis data SQL. Tabel berikut menunjukkan hasil pengujian trasnformasi seluruh isi ontologi. Dari tabel ini, jika dibandingkan dengan tabel informasi, dapat dilihat bahwa aplikasi OntoDB telah berhasil mentransformasi seluruh isi ontologi dengan sempurna.

Tabel 3. Hasil Pengujian Transformasi Seluruh Isi Ontologi

Hal	University Ontology	Pizza Ontology	Kennedys Ontology
Jumlah Kelas	43	99	9
Jumlah Object Properti	30	8	16
Jumlah Datatype Properti	7	0	8
Jumlah *Instance*	1657	5	135
Jumlah Kelas Tanpa Nama	14	232	2
Jumlah Kelas Intersection Tanpa Nama	6	15	1
Jumlah Enumeration Tanpa Nama	0	1	0
Jumlah Kelas Complement Tanpa Nama	0	3	0
Jumlah Kelas Restriction	8	188	0
Jumlah Hubungan Equivalent Class	0	15	2
Jumlah Hubungan Intersection	0	0	0
Jumlah Hubungan Disjoint With	0	796	0
Jumlah Hubungan Enumeration	0	1	0
Jumlah Hubungan SubClassOf	38	259	10
Jumlah Hubungan SubPropertyOf	5	4	8
Jumlah Hubungan InverseOf	3	6	1
Jumlah Hubungan Equivalent Properti	0	0	1
Jumlah Hubungan Different From	0	0	0
Jumlah Hubungan Same As	0	0	2
Jumlah Hubungan Imported Ontology	1	0	0

4.2 Hasil Pengujian Mentransformasi Sebagian Isi Ontologi

Pengujian ini dijalankan dengan menggunakan file ontologi University Ontology. Pengujian dijalankan dengan memilih beberapa kelas, properti, dan *instance*. Tabel berikut menunjukkan teknik pengujian yang dijalankan terhadap file ontologi tersebut.

Tabel 2 Pengujian 1 Mentransformasi dua kelas saja

Pengujian 1	Menguji kemampuan mentransformasi dua kelas saja
Kelas yang dipilih	Research Assistant dan Undergraduate Student
Properti yang dipilih	Semua object properti dan datatype properti
Instance yang dipilih	Semua *instance* dari Research Assistent dan Undergraduate Student
Hasil pengujian	Semua kelas, properti dan *instance* yang dipilih telah berhasil ditransformasikan ke kode SQL

Tabel 3 Pengujian 2 Mentransformasi dua kelas dan datatype properti saja

Pengujian 2	Menguji kemampuan mentransformasi dua kelas dan datatype properti saja
Kelas yang dipilih	Research Assistant dan Undergraduate Student
Properti yang dipilih	Datatype properti
Instance yang dipilih	Semua *instance* dari Research Assistent dan Undergraduate Student
Hasil pengujian	Semua kelas, properti dan *instance* yang dipilih telah berhasil ditransformasikan ke kode SQL

Tabel 4 Pengujian 3 Mentransformasi beberapa *instance* saja

Pengujian 3	Menguji kemampuan mentransformasi beberapa *instance* saja
Kelas yang dipilih	Research Assistant dan Undergraduate Student
Properti yang dipilih	Semua object properti dan datatype properti
Instance yang dipilih	GraduateStudent104, GraduateStudent105, UndergraduateStudent104, UndergraduateStudent105
Hasil pengujian	Semua kelas, properti dan *instance* yang dipilih telah berhasil ditransformasikan ke kode SQL

Tabel 5 Pengujian 4 Mentransformasi kelas dan subkelasnya

Pengujian 4	**Menguji kemampuan mentransformasi kelas dan subkelasnya**
Kelas yang dipilih	Work dan semua subkelasnya yaitu Course, GraduateCourse, dan Research
Properti yang dipilih	listedCourse, takesCourse
Instance yang dipilih	Semua *instance* dari Work dan subkelas Work
Hasil pengujian	Semua kelas, properti dan *instance* yang dipilih telah berhasil ditransformasikan ke kode SQL

Dari hasil pengujian tersebut, aplikasi OntoDB dapat mentransformasi sebagian saja isi dari ontologi dengan baik. File SQL yang dihasilkan dari transformasi sebagian isi ontologi ini dapat di-*import* ke dalam basis data MySQL.

5.Kesimpulan

Aplikasi Pemetaan Ontologi kepada Basis Data (OntoDB) ini telah diimplementasikan sesuai dengan desain yang telah dirancang berdasarkan analisis yang telah dibuat sebelumnya. Aplikasi OntoDB ini telah berhasil dikembangkan dan menghasilkan aplikasi yang dapat memaparkan isi file ontologi dan memetakan ontologi kepada kode basis data SQL. Aplikasi ini memiliki lima fitur utama yaitu:

a. Membuka File Ontologi
b. Memilih Kelas
c. Memilih Properti
d. Memilih *Instance*
e. Menghasilkan file berisi kode SQL

Aplikasi Pemetaan Ontologi kepada Basis Data (OntoDB) ini memiliki beberapa kelebihan dibandingkan dengan aplikasi lain yaitu:

a. Dapat membatasi pengguna dari membuka file selain file OWL, RDFS, dan RDF
b. Memaparkan isi kelas dan properti dalam bentuk struktur pohon (tree-structure)
c. Antarmuka yang mudah untuk dimengerti oleh pengguna
d. Membolehkan pengguna memetakan sebagian saja dari isi ontologi ke SQL

Aplikasi OntoDB dapat dikembangkan lagi dengan menambahkan beberapa fitur berikut:

a. Menambahkan jenis file ontologi yang data dibuka seperti format TURTLE dan N-Triple.
b. Menambahkan fungsi untuk membuka dan memetakan beberapa ontologi sekaligus.
c. Menambahkan fungsi menggabungkan hasil pemetaan beberapa ontologi kepada satu file SQL yang sama.

Referensi

[1] T. Berners-Lee, J. Hendler, and O. Lassila. 2001. *The Semantic Web*. Sci. Am. 284 (5): 34–43.
[2] I. Astrova, N. Korda, and A. Kalja. 2007. *Storing OWL Ontologies in SQL Relational Databases*. Eng. Technol. 1 (4):167–172.
[3] A. Khalid, S. A. H. Shah, and M. A. Qadir. 2009. *OntRel: An Ontology Indexer to Store OWL-DL Ontologies and Its Instance*. Int. Conf. of Soft. Comp. and Pattern Recognition. 478–483.
[4] A. Gali, C. X. Chen, K. T. Claypool, and R. Uceda-Sosa. 2004. From Ontology to Relational Databases. In: Wang S. et.al. (eds) Conceptual Modeling for Advanced Application Domains. Lecture Notes in Computer Science. 3289: 278–289.
[5] E. Vysniauskas and L. Nemuraite. *Transforming ontology representation from OWL to relational database*. Inf. Technol. Control. 35 (3): 333–343.
[6] H. Afzal. 2016. *OWLMap : Fully Automatic Mapping of Ontology into Relational Database Schema*. Int. J. of Adv. Comp. Sci. and Apps.7 (11): 7–15.
[7] H. Zhang, Z. Wang, Z. Gao, and W. Li. 2009. *Design and Implementation of Mapping Rules from OWL to Relational Database.* 2009 WRI World Congress on Comp. Sci. and Inf. Eng. 4: 71–75.
[8] R. S. Pressman. 2005. *Software Engineering: A Practitioner's Approach 6th ed*. New York, NY, USA: McGraw-Hill, Inc.