**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Faiq**

**NRP : 5115100007**

**DOSEN PEMBIMBING : 1. Nurul Fajrin Ariyani, S.Kom, M.Sc.  
 2. -**

# JUDUL TUGAS AKHIR

“Rancang Bangun Aplikasi Berbasis Web untuk Visualisasi *Family Tree* Tokoh Sejarah Indonesia Menggunakan Ontologi”

# LATAR BELAKANG

Tokoh bersejarah adalah seseorang yang namanya dikenang karena jasanya. Sedangkan pahlawan adalah gelar penghargaan yang diberikan kepada seseorang atas tindakan heroiknya. Tokoh bersejarah dan pahlawan, keduanya menjadi bukti dari adanya suatu kejadian penting di masa lalu. Berbagai macam cerita dan kejadian yang menjadi kenangan sejarah sepanjang masa melibatkan nama tokoh-tokoh penting tersebut. Indonesia memiliki banyak catatan nama tokoh-tokoh bersejarah dan pahlawan nasional. Setiap tokoh memiliki kisah serta rekan hidup yang berbeda. Rekan hidup dapat berarti keluarga, sahabat, teman, dan sebagainya. Berdasarkan pada *history* rekan hidup, tokoh yang satu dengan tokoh yang lain memiliki hubungan terkait sehingga relasi antar tokoh tersebut dapat diketahui. Selain itu, hubungan tersebut juga dapat menentukan kejadian apa yang pernah terlibat di antara mereka.

Dalam perkembangan teknologi, pengetahuan tentang tokoh bersejarah dan pahlawan nasional tidak hanya terhimpun di dalam buku-buku sejarah. Banyak situs daring yang menyediakan informasi tentang tokoh bersejarah dan pahlawan nasional, seperti Wikipedia, DBpedia, Everything2, Quora, dan lain-lain. Akan tetapi dalam situs-situs tersebut, mayoritas informasi yang diberikan masih berupa paragraf-paragraf teks. Otak manusia dapat memproses informasi visual 60.000 kali lebih cepat daripada informasi teks, oleh karena itu, untuk memudahkan pemahaman terkait tokoh bersejarah dan relasinya, diperlukan sebuah aplikasi yang menampilkan silsilah keluarga tokoh bersejarah berbasis web.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana menentukan *family* tokoh sejarah Indonesia dengan rekan hidupnya?
2. Bagaimana menentukan *query* yang mendefinisikan batas-batas relasi dalam *family tree* tokoh sejarah Indonesia?
3. Bagaimana membuat aplikasi untuk menampilkan visualisasi *family tree* tokoh?

# BATASAN MASALAH

Permasalahan yang dibahasa dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Agen *person* yang digunakan adalah tokoh-tokoh sejarah Indonesia.
2. Jumlah data *test case* yang digunakan adalah 10 tokoh bersejarah.
3. Data bersumber dari informasi yang dimuat Wikipedia dan DBpedia mengenai tokoh sejarah Indonesia.
4. Format file yang digunakan adalah JSON.
5. Pengembangan web menggunakan framework Laravel.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pengerjaan tugas akhir ini adalah membuat aplikasi web sederhana yang dapat menampilkan silsilah keluarga dari tokoh sejarah Indonesia secara visual dari file JSON yang didapat dari DBPedia.

# MANFAAT TUGAS AKHIR

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Mempermudah pemahaman informasi mengenai silsilah keluarga tokoh sejarah Indonesia.
2. Menambah wawasan untuk mengenal tokoh-tokoh lain yang masih ada relasi dengan seorang tokoh sejarah Indonesia tertentu.

# TINJAUAN PUSTAKA

* 1. **Tokoh Bersejarah**

Di Indonesia, tokoh bersejarah biasanya tokoh yang dianggap penting dalam proses terbentuknya Indonesia sampai sekarang.

* 1. **Ontologi**

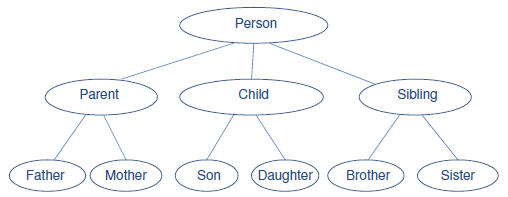
Istilah ontologi awalnya berasal dari lingkup ilmu filsafat yang kemudian dikooptasi oleh ilmu komputer. Ontologi didefinisikan sebagai studi tentang sebuah konsep yang secara sistematik menjelaskan tentang segala sesuatu yang ada atau nyata. Terdapat beberapa definisi mengenai ontologi diantaranya:

* “Spesifikasi eksplisit dan formal dari sebuah konseptualisasi.”
* “Mendefinisikan konsep dan hubungan yang digunakan untuk mendeskripsikan serta menunjukkan pengetahuan tertentu.” (W3C)
* Model formal dari hal-hal yang ada dalam domain pengetahuan tertentu dan hubungan di antara hal-hal tersebut. Hal yang dimaksud dapat berarti konsep, pekerjaan, orang, tempat, objek, peristiwa, dll.

Terdapat 3 komponen utama ontologi yaitu *class, property*, dan i*nstance*. Berikut adalah penjelasan mengenai komponen-komponen tersebut:

* *Class*

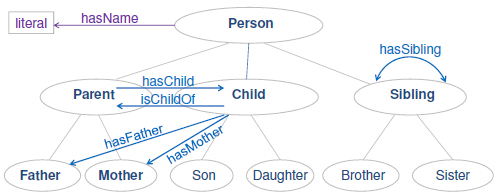
*Class* menspesifikasikan *property* yang sama dari beberapa *instance* dan mendukung klasifikasi hierarkikal. Selain itu, class juga mencakup *superclass* dan *subclass. Subclass* merupakan versi yang lebih khusus dari *superclass*nya. Setiap *subclass* mewarisi operasi dan atribut dari leluhurnya. *Subclass* mungkin memiliki operasi dan atributnya sendiri (yang tidak dimiliki oleh leluhurnya). Hubungan antara *subclass* dan *superclass* digambarkan dengan *class hierarchy* yang dicontohkan pada **Gambar 1**.



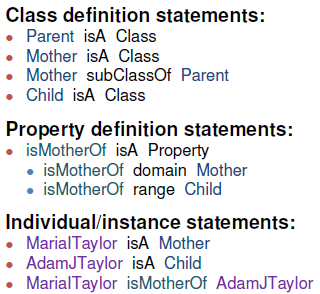
Gambar 1. Class Hierarchy

* *Property*

Dalam konteks ontologi, *property* menghubungankan *member* dari suatu kelas ke *member* kelas lainnya. *Property* mencakup *subproperties* dan *superproperties*. *Property* juga disebut sebagai “slots” dalam terminologi yang lain. Skema *property* dicontohkan pada **Gambar 2**.



Gambar 2. Property

* *Instance*

*Instance* merujuk pada sesuatu yang teridentifikasi dalam model konseptual. *Instance* juga disebut sebagai individual. Sehingga dapat dikatakan bahwa *instance* menspesifikasikan *member* dari suatu *class*. *Instance* dapat berwujud (alat, bagian, benda, dkk) atau tidak berwujud (sebuah konsep, konstruksi mental). Contoh relasi antara *class, property*, dan *instance* dapat dilihat pada **Gambar 3**.

Gambar 3. Contoh Ontologi

Selain 3 komponen utama di atas, terdapat beberapa istilah yang perlu dipahami dalam konteks ontologi antara lain :

* *Inheritance* yang menurunkan atribut atau perilaku dalam sebuah kelas objek.
* *Domain* yang merupakan member dari suatu kelas yang dapat menjadi subjek dari *property* yang diberikan.
* *Range* yang merupakan member dari suatu kelas yang dapat menjadi objek dari *property* yang diberikan.
* *Constraint* dan *rule* yang menentukan batasan dan istilah-istilah teknis untuk mendukung *reasoning*.
* *Relationship* yang merupakan mekanisme inferensi untuk menggenerasi pengetahuan baru.

Dalam *semantic modelling*, ontologi direpresentasikan dengan bahasa yang terstandarisasi yaitu RDF, RDFS, atau OWL. Kegunaan ontologi secara umum adalah sebagai *controlled vocabulary, semantic interoperability, knowledge sharing,* dan *reuse*. Selanjutnya proses pengembangan sebuah model ontologi dapat dilakukan dengan tahapan sebagai berikut:

* Tahap penentuan domain.
* Tahap penggunaan ulang.
* Tahap penentuan istilah pada ontologi.
* Tahap pendefinisian *class* dan *hierarchy class.*
* Tahap pendefinisian *properties.*
* Tahap pendefinisian *constraints.*
* Tahap pembuatan *instance.*
  1. **OWL**

OWL (*Web Ontology Language*) adalah bahasa yang digunakan untuk merepresentasikan makna dari kosakata dan relasi antar kata sehingga makna suatu informasi menjadi eksplisit. OWL merupakan ekstensi daei RDF Schema. W3C *(World Wide Web Consortium)* merekomendasikan OWL dalam penulisan ontologi untuk web semantik. OWL menggunakan kemampuan RDF untuk memodelkan informasi dari sekumpulan data. Sedangkan data model merupakan suatu tata bahasa yang digunakan untuk mengartikan sekumpulan data untuk menjadi suatu informasi.

Elemen pada OWL terdiri atas *classes, properties, instances of classes,* dan relasi antar *instances*.

* ***Classes***

OWL mendefinisikan *root* dari semua yang ada dengan owl:Thing. Sehingga semua *class* yang dibuat secara implisit merupakan *subclass* owl:Thing. Pembuatan *class* menggunakan owl:Class dan menyatakan *subclass* dengan rdfs:subClassOf.

*Class* memiliki beberapa deskripsi, antara lain:

1. ***Equivalent classes***

|  |
| --- |
| ***Parent*** *it is any* ***Person*** *who has a child.* |

1. ***Superclasses***
2. ***Members***

Menjelaskan individual yang merupakan *member* dari *class* yang dirujuk.

1. ***Disjoint classes***

*Member* dari *class* tersebut tidak bisa menjadi *member* di *class* yang lain. Biasanya tidak perlu didefinisikan, kecuali jika menggunakan *external reasoning* atau aplikasi lain yang membutuhkan *class* definisi yang lebih eksplisit. Misal:

|  |
| --- |
| ***Father*** *disjoint class with* ***Mother*** |

Konsep yang paling dasar dalam sebuah domain harus sesuai dengan *class* yang menjadi *root* dari berbagai taksonomi. Contohnya untuk domain anggur, diciptakan tiga kelas root: Winery, Region, dan ConsumableThing sebagai berikut:

|  |
| --- |
| <owl:Class rdf:ID="Winery"/>  <owl:Class rdf:ID="Region"/>  <owl:Class rdf:ID="ConsumableThing"/> |

* ***Individuals***

*Individuals* atau disebut juga *instances* adalah anggota dari *classes*. *Instances* dapat dipandang sebagai objek yang ada pada domain tertentu. Sama seperti owl:Class yang menjadi *meta level* untuk *class*, begitu juga *class* yang telah didefinisikan menjadi *mata level* untuk *instance*. Sebuah individu diperkenalkan dengan mendeklarasikan individu tersebut untuk menjadi *member* dari sebuah *class*.

|  |
| --- |
| <Region rdf:ID="CentralCoastRegion" /> |

Selain dideklarasikan seperti contoh di atas, pendeklarasian individu juga dapat dilakukan dengan *syntax* yang identik sebagai berikut:

|  |
| --- |
| <owl:Thing rdf:ID="CentralCoastRegion" />  <owl:Thing rdf:about="#CentralCoastRegion">  <rdf:type rdf:resource="#Region"/>  </owl:Thing> |

* ***Properties***

*Property* merupakan relasi *binary*. Ada dua jenis *property* pada OWL, yaitu ObjectProperty (relasi antara instance dari dua *classes*) dan DatatypeProperty (relasi antara *instance* dengan RDF literal dan XML Schema *datatype*). Sama halnya seperti *class* yang dapat dinyatakan sebagai subPropertyOf dengan rdfs:subPropertyOf. Untuk memberikan batasan pada suatu *property*, dapat digunakan rdfs:domain dan rdfs:range, yang disebut juga sebagai *global restriction* karena berlaku untuk umum, tidak terbatas pada *class* tertentu.

**DatatypeProperty** yang dapat digunakan adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| string | normalizedString | boolean |
| decimal | float | double |
| integer | nonNegativeInteger | positiveInteger |
| nonPositiveInteger | negativeInteger |  |
| long | int | short | byte |
| unsignedLong | unsignedInt | unsignedShort | unsignedByte |
| hexBinary | base64Binary |  |  |
| dateTime | time | date | gYearMonth |
| gYear | gMonthDay | gDay | gMonth |
| anyURI | token | language |  |
| NMTOKEN | Name | NCName |  |

Terdapat beberapa mekanisme yang digunakan untuk penentuan *property*. Tidak menutup kemungkinan bahwa mekanisme tersebut memanfaatkan *object* *property characteristics* yang menggunakan penalaran. ***Object* *property characteristics*** yang dapat digunakan adalah sebagai berikut:

1. ***Functional and Inverse functional***

*Functional property* hanya boleh memiliki satu *member* di *range*nya untuk semua *member domain*.

Contoh: hasGender

1. ***Transitive***

|  |
| --- |
| *(x property y) and (y property z) implies (x property z)* |

Contoh: hasSibling

1. ***Symmetric***

|  |
| --- |
| *if (x property y) then (y property x)* |

Contoh: hasSpouse

1. ***Asymmetric***

|  |
| --- |
| *if (x property y) then (y property x) → false* |

Contoh: hasWife

1. ***Reflexive***

Individu bisa berelasi dengan dirinya sendiri.

1. ***Irreflexive***

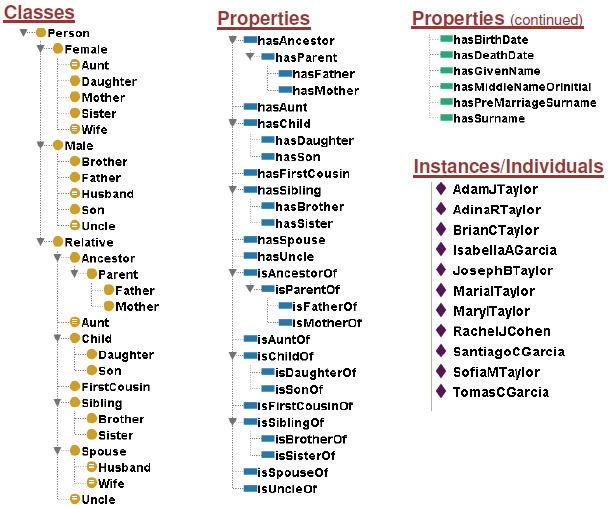
Tidak boleh ada individual yang berelasi dengan dirinya sendiri untuk *property* tersebut.

Contoh: isParentOf

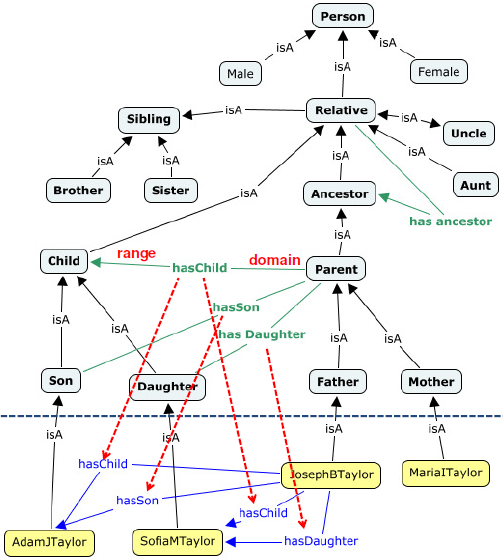
Dalam proses *reasoning* pada OWL, dapat digunakan beberapa *reasoner* seperti Pellet, FACT++, HermiT, dan lain sebagainya. *Reasoner* ini digunakan untuk memeriksa konsistensi pada ontologi, melakukan klarifikasi secara otomatis berdasarkan relasi hierarki *(subsumption reasoning)*, dan mendapatkan data atau fakta baru berdasarkan *axioms* dan *rules*. Terdapat beberapa contoh penggunaan ontologi, khususnya OWL.

* *Web portal*
* *Multimedia collectionns*
* *Corporate web site management*
* *Design documentation*
* *Agents and services*
* *Ubiquitous computing*
  1. **Family Relationships Ontology**

Hubungan dalam sebuah keluarga kini dapat diketahui dengan mudah melalui *Family Relationships Ontology*. Ontologi ini memiliki beberapa kelebihan, diantaranya adalah dapat diketahuinya keakraban, relasi, warisan, *domain, range, property, constraint,* dan kesimpulan logis dalam sebuah keluarga secara praktis. *Family Relationships Ontology* juga mempermudah untuk memahami ontologi dan individu. Selain itu, hubungan kekeluargaan dapat digeneralisasikan ke domain pengetahuan lainnya karena konsepnya yang mudah dipahami. **Gambar 4** menunjukkan spesifikasi dari *Family Relationships Ontology.* Sedangkan *class hierarchy* dari spesifikasi tersebut dapat dilihat pada **Gambar 5**.



Gambar 4. Family Relationships Ontology



Gambar 5. Family Relationships Map

* 1. **FOAF (Friend of a Friend)**

FOAF adalah mesin pembaca ontologi yang menspesifikasikan orang, kegiatan, dan hubungan mereka dengan orang atau hal-hal lain. Pada dasarnya, FOAF adalah proyek yang ditujukan untuk menghubungkan orang dan informasi menggunakan web. Tanpa memperhatikan informasi yang terdapat dalam pikiran manusia, informasi dalam bentuk digital atau dalam bentuk data faktual dapat dihubungkan. FOAF mengintegrasikan tiga jenis jaringan: jaringan sosial antar manusia, persahabatan dan asosiasi, dan jaringan representasional yang menggambarkan konsep sederhana dari fakta alam semesta.

FOAF memiliki 3 hal utama yang dikelompokkan dalam kategori.

* *Core*

*Class* dan *property* yang membentuk inti dari FOAF. Dalam hal ini, karakteristik dari orang atau kelompok sosial yang tidak terikat dengan waktu dan teknologi digambarkan. Dengan demikian, didapatkan gambaran informasi tentang sejarah, warisan budaya, dan konteks *digital library* mengenai orang tersebut. Kategori yang terdapat dalam *Core* dapat dilihat pada **Tabel 1**.

Tabel 1. FOAF Core

|  |  |  |
| --- | --- | --- |
| **FOAF Core** | | |
| Agent  Person  name  title  img  depiction (depicts)  familyName  givenName  knows  based\_near  age  made (maker)  primaryTopic (primaryTopicOf) | Project  Organization  Group  member | Document  Image |

* *Social Web*

Selain bagian inti dari FOAF, terdapat sejumlah istilah yang digunakan ketika menggambarkan akun internet, *addressbooks*, dan aktivitas lain yang berkaitan dengan web. Kategori yang terdapat dalam *Social Web* dapat dilihat pada **Tabel 2.**

Tabel 2. FOAF Social Web

|  |  |  |
| --- | --- | --- |
| **FOAF Social Web** | | |
| nick  mbox  homepage  weblog  openid  jabberID  mbox\_sha1sum  interest  topic\_interest | topic (page)  workplaceHomepage  workInfoHomepage  schoolHomepage  publications  currentProject  pastProject  account | OnlineAccount  accountName  accountServiceHomepage  PersonalProfileDocument  tipjar  sha1  thumbnail  logo |

* *Linked Data Utilities*

FOAF dimulai sebagai proyek RDFWeb dan membangun sebuah model yang banyak digunakan untuk mempublikasikan data sederhana melalui keterkaitan antar dokumen RDF. FOAF tetap berperan penting untuk pengembangan “Linked Data”, tetapi juga tetap mempertahankan *scope* untuk informasi yang tidak mudah diringkas sebagai data faktual sederhana.

Awalnya, FOAF adalah web yang digunakan untuk mengintegrasikan informasi faktual dengan informasi dalam bentuk dokumen *human-oriented* (misalnya: video, buku, *spreadsheet*, 3D model), serta informasi yang hanya terekam dalam pikiran manusia. Atas dasar itulah FOAF merupakan 'percobaan' yang digunakan untuk memenuhi berbagai tujuan besar pendidikan (Misalnya: Geekcode) dan juga beberapa istilah utilitas teknis (Misalnya: *focus*, *LabelProperty*) yang mendukung hubungan antar informasi. Istilah kata yang lebih luas memungkinkan FOAF untuk menggunakan pragmatisme tertentu seperti seperangkat istilah yang berguna untuk *Web Community*, sekaligus untuk menekankan ide utama FOAF, yaitu menghubungkan *networks of information* dengan *networks of people*.

Selanjutnya, *class* dan *property* yang dimiliki oleh FOAF dapat dilihat pada **Tabel 3**.

Tabel 3. FOAF Classes & Properties

|  |  |  |
| --- | --- | --- |
| **Classes** | **Properties** | |
| Agent  Document  Group  Image  LabelProperty  OnlineAccount  OnlineChatAccount  OnlineEcommerceAccount  OnlineGamingAccount  Organization  Person  PersonalProfileDocument  Project | account  accountName  accountServiceHomepage  age  aimChatID  based\_near  birthday  currentProject  depiction  depicts  dnaChecksum  familyName  family\_name  firstName  focus  fundedBy  geekcode  gender  givenName  givenname  holdsAccount  homepage  icqChatID  img  interest  isPrimaryTopicOf  jabberID  knows  lastName  logo  made | maker  mbox  mbox\_sha1sum  member  membershipClass  msnChatID  myersBriggs  name  nick  openid  page  pastProject  phone  plan  primaryTopic  publications  schoolHomepage  sha1  skypeID  status  surname  theme  thumbnail  tipjar  title  topic  topic\_interest  weblog  workInfoHomepage  workplaceHomepage  yahooChatID |

* 1. **BIO**

BIO adalah sebuah dokumen yang memuat kosakata untuk menggambarkan informasi biografis seseorang, baik yang masih hidup atau sudah meninggal. BIO Vocabulary berisi hal berguna untuk mengetahui lebih banyak tentang latar belakang seseorang dan informasi hubungan silsilahnya. Pendekatan ini dilakukan dengan menggambarkan kehidupan seseorang sebagai rangkaian peristiwa penting yang saling berhubungan dengan informasi lainnya yang dapat berkaitan. BIO Vocabulary mendefinisikan *event framework* dan menyediakan jenis-jenis *event* yang mencakup banyak kasus penggunaan. Namun kedepannya, diharapkan BIO Vocabulary akan dikembangkan dengan *vocabulary* lain untuk memenuhi berbagai kebutuhan yang ingin dicapai. Tujuan dari BIO Vocabulary ini adalah untuk menggambarkan peristiwa biografi orang dan mendefinisikan *property* serta *class* seseorang yang terlibat dalam suatu *event*.

Pada intinya BIO Vocabulary berkaitan dengan orang, hubungan mereka dan peristiwa-peristiwa dalam hidup mereka. Kaitan tersebut dapat digunakan untuk membangun sebuah narasi kehidupan seseorang dan interaksi mereka dengan orang lain atau organisasi serta dunia di sekitar mereka. Interval waktu juga mungkin terikat dengan hubungan khusus antara orang-orang dan kelompok-kelompok atau organisasi. Berbagai jenis *event* dalam kehidupan didefinisikan melalui *vocabulary* ini, termasuk kelahiran, pernikahan dan kematian. Selain itu, ada juga penobatan, kinerja dan bahkan pembunuhan. Peristiwa ini tidak dimaksudkan untuk menjadi sesuatu yang komprehensif, tetapi merupakan sampel dari jenis *event* yang berkaitan dengan area biografi. Saat ini, jenis hubungan antar kosakata hanya disediakan oleh *Relationship Class* tertentu. Hal ini terjadi karena pertimbangan jenis hubungan seperti keluarga, kepegawaian, dan kepemilikan akan ditentukan di masa depan. Urutan kejadian dan interval waktu adalah penyusun *timeline* sejarah dimana orang dan hubungan mereka dapat ditempatkan. *Event* yang diurutkan dalam waktu akan mengaitkan satu sama lain dan mengabstraksikan interval waktunya. Kategori properties dan event yang dimiliki oleh BIO dapat dilihat pada **Tabel 4**.

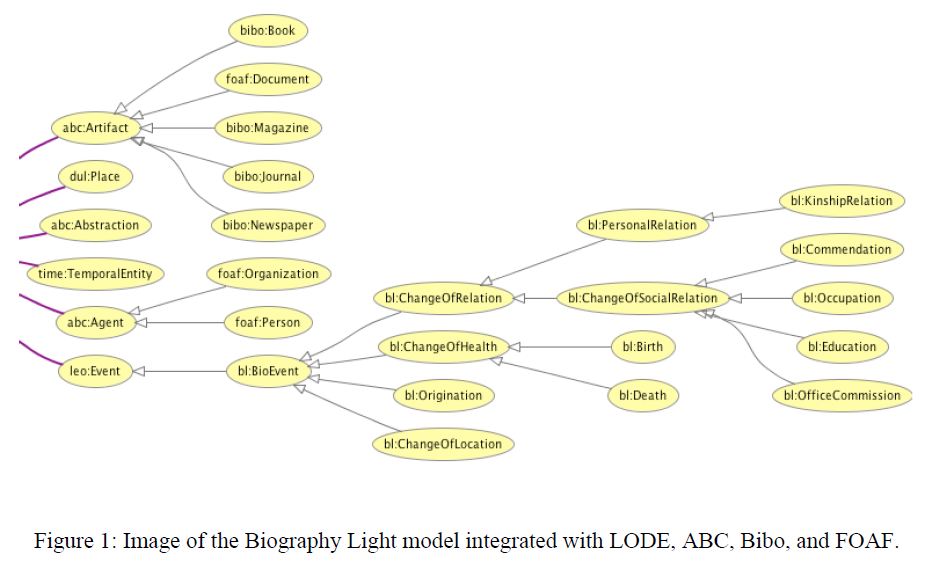
Tabel 4. Bio Terms

|  |  |  |
| --- | --- | --- |
| **Properties of a person** | One-line bio  Biography  Key Words  Father  Mother | Child  Life Event  Birth Event  Death Event |
| **Types of event** | Event  Individual Event  Group Event  Accession  Adoption  Annulment  Assassination  Baptism  BarMitzvah  BasMitzvah  Birth  Burial  Change of Position  Coronation  Cremation  Death  Demotion  Dismissal  Divorce | Emigration  Employment  Execution  Enrolment  Funeral  Graduation  Imprisonment  Inauguration  Investiture  Marriage  Murder  Change of Name  Naturalization  Ordination  Promotion  Redundancy  Resignation  Retirement |
| **Properties of an event** | Date  Place | State  Position |
| **Properties that relate an event to an agent** | Agent  Parent  Employer  Officiator  Organization | Principal  Partner  Witness  Spectator |
| **Properties that relate an event to another event** | Concurrent Event  Following Event  Preceding Event | Immediately Following Event  Immediately Preceding Event |

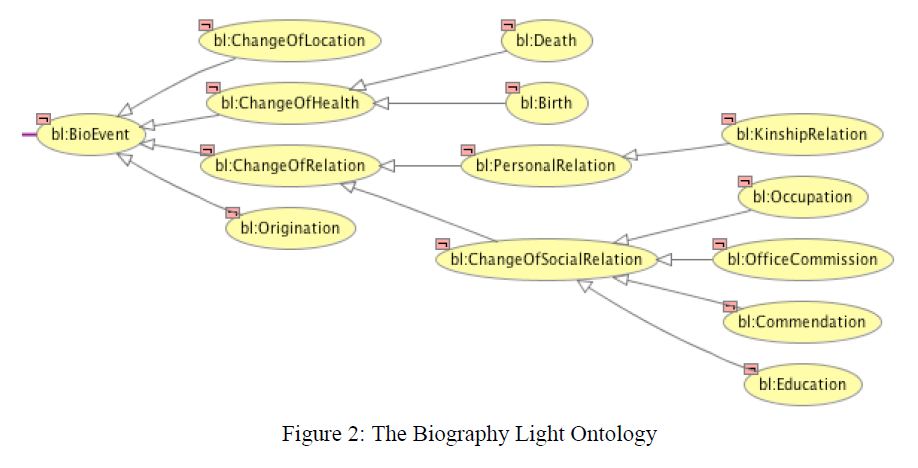
* 1. **Biography Light Ontology**

*Biography Light Ontology* adalah hasil dari dua proposal penelitian oleh The Electronic Cultural Atlas Initiative, the Bringing Lives to Light: *Biography in Context project and the Support for the Learner: What, Where, When, and Who*. Tujuan dari penelitian ini adalah untuk merancang dan memvalidasi standar serta praktik terbaik dalam pengkodean peristiwa atau aktivitas kehidupan berdasar pada model heuristik apa, dimana, kapan, dan siapa.

Model *Biogrpahy Light* dibentuk dari ontologi RDF + OWL dalam upaya untuk meningkatkan konektivitas dan interoperabilitas antar standar *encoding* teks biografis yang sudah ada. Ontologi yang digunakan dalam penelitian ini meliputi *ABC Ontology*, *Linking Open Description of Events ontology* (LODE), *Friend of a Friend* (FOAF), *Bio vocabulary, Relationship vocabulary,* dan *Bibliographic Ontology* (Bibo). Integrasi antara model *Biography Light* dengan LODE, ABC, Bibo, dan FOAF ditunjukkan pada **Gambar 6**. Sedangkan hirarki dari *Biography Light Ontology* dapat dilihat pada **Gambar 7**.



Gambar 6. Model Biography Light yang diintegrasikan dengan LODE, ABC, Bibo, and FOAF

****

Gambar 7. Biography Light Ontology

* 1. **SPARQL**

SPARQL (diucapkan "sparkle", sebuah akronim untuk *SPARQL Protocol and RDF Query Language*) merupakan bahasa kueri untuk RDF. Dalam kuerinya, SPARQL menggunakan pola *triple*. Pola triple menyerupai *triple* RDF tetapi subjek, predikat, dan objeknya dapat berupa variabel.

Ada empat jenis kueri SPARQL, yaitu SELECT, CONSTRUCT, ASK, DESCRIBE. Seseorang dapat menjalankan kueri SPARQL melalui SPARQL Endpoint.

* 1. **Jena API**

Apache Jena adalah Java *framework* untuk *open source Semantic Web*. Jena menyediakan API untuk mengekstrak data dan menulisnya dalam grafik RDF. Grafik direpresentasikan sebagai abstrak "Model". Sebuah Model dapat bersumber pada data yang berasal dari file, database, URL atau kombinasi dari ketiganya. Sebuah Model juga dapat dikueri menggunakan SPARQL 1.1.

Jena mirip dengan ‘Sesame’, meskipun tidak sama seperti Sesame. Jena dapat digunakan pada OWL. *Framework* ini memiliki berbagai *internal reasoner* dan Pellet *reasoner* yang dapat diatur untuk berfungsi di Jena.

* 1. **Pellet Reasoner**

Pellet didasarkan pada algoritma tableaux yang dikembangkan untuk mengekspresikan *Description Logics*. Pellet mendukung semua konstruksi OWL DL termasuk owl:oneOf dan owl: hasValue. Saat ini, belum terdengar adanya algoritma lengkap yang *decidable* dan efektif untuk semua OWL DL (khususnya, penanganan *inverse properties* dan *cardinality restrictions*). Pellet mengkombinasikan algoritma yang lengkap sebagai reasoner, yaitu OWL DL tanpa *nominals* (SHIN (D)) dan OWL DL tanpa *invers properties* (SHON (D)). Algoritma ini dikombinasikan untuk mendapatkan penalaran yang lengkap dan berkaitan dengan semua DL. Pellet telah terbukti praktis berguna dalam berbagai pekerjaan saat ini. **Gambar 8** menunjukkan komponen utama Pellet *reasoner*.



Gambar 8. Arsitektur Pellet Reasoner

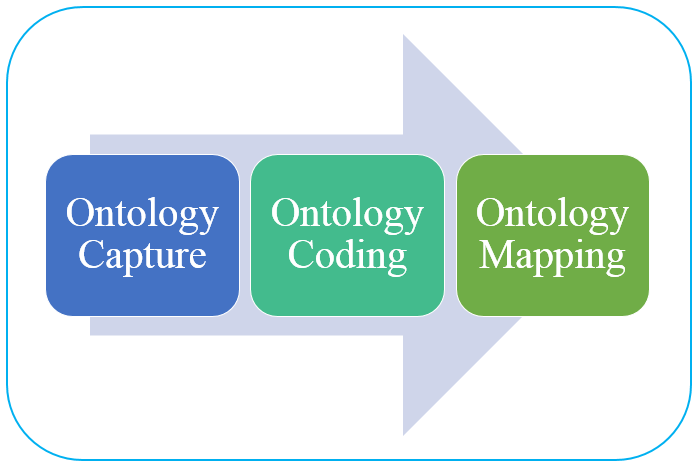
Ontologi OWL diparsing ke dalam RDF pola triple (RDF / XML, N3 dan N-Triple syntaxes yang mendukung). Pellet memvalidasi spesies ontologi dimana *triple* *RDF* dikonversi menjadi pernyataan dan aksioma berbasis pengetahuan. Jika level ontologi adalah OWL Full karena hilangnya tipepola *triple*, maka Pellet menggunakan beberapa heuristik untuk memperbaiki ontologi. Misalnya *untyped resource* yang telah digunakan dalam predikat *position* dalam sebuah pola *triple* akan disimpulkan menjadi *datatype property* jika *triple* literal dalam posisi objek.

Pellet menyimpan aksioma tentang kelas-kelas dalam komponen TBox dan menyimpan pernyataan tentang individu dalam komponen abox. Partisi TBox, adalah tempat penyerapan dan optimasi berlangsung. Tableau reasoner menggunakan *rule* tableau standar dan mencakup berbagai optimasi standar seperti ketergantungan yang diarahkan pada *backjumping*, percabangan semantik dan strategi pemblokiran awal. *Datatype reasoning* untuk *built-in* dan pengambilan *XML Schema datatypes* primitif didukung dalam *reasoner* ini. Pellet diimplementasikan dalam Java dan berada di bawah lisensi MIT. Sumber file bersama dengan beberapa dokumentasinya dapat didownload dari halaman PelletWeb (http://www.mindswap.org/2003/pellet/).

# RINGKASAN ISI TUGAS AKHIR

*Cultural heritage* (Keragaman budaya) dapat terlahir dari tokoh-tokoh penting seperti tokoh bersejarah dan pahlawan. Indonesia memiliki banyak tokoh bersejarah dan pahlawan nasional. Setiap tokoh memiliki biografi hidup yang berbeda. Biografi mencatat keterkaitan antar seorang tokoh dengan yang lainnya. Keterkaitan tersebut dapat digambarkan dengan ontologi. *Actor* adalah salah satu domain dalam *cultural heritage* yang dapat diontologikan. Ruang lingkup *actor* mencakup *person, group,* dan *organization*. Sampel dari *person* adalah tokoh-tokoh penting yang telah disebutkan sebelumnya, yaitu tokoh bersejarah dan pahlawan nasional.

Tujuan dari pengerjaan tugas akhir ini adalah menghasilkan suatu metode ontologi yang dapat melakukan pencarian relasi antar tokoh dalam domain sejarah Indonesia secara otomatis. Untuk mencari hubungan antar tokoh sejarah Indonesia, perlu mengembangkan beberapa ontologi seperti *Family Relationships Ontology*, FOAF (*Friend of a Friend*), BIO, dan *Biography Light Ontology*. Pengembangan ontologi tersebut dapat dilakukan dengan *Ontology Web Language* (OWL) menggunakan *tools* Protege. Tahapan-tahapan yang akan dilakukan untuk mengembangkan ontologi ditunjukkan pada **Gambar 9**.



Gambar 9. Tahapan Pengembangan Ontologi

* **Ontology Capture**

Proses mengidentifikasi dan mendefinisikan konsep serta istilah-istilah penting. Adapun langkah-langkah untuk melakukan *ontology capture* adalah sebagai berikut:

1. Mengidentifikasi inti konsep dan hubungan dari domain yang difokuskan.
2. Mendefinisikan teks-teks ambigu untuk konsep dan hubungan tersebut dengan tepat .
3. Mengidentifikasi istilah untuk menyebut konsep dan hubungan tersebut.
4. Memastikan kembali istilah yang telah dipilih

* **Ontology Coding**

Proses menuliskan ontologi dalam bahasa pengkodean formal. Untuk melakukan *ontology coding*, berikut adalah langkah-langkahnya:

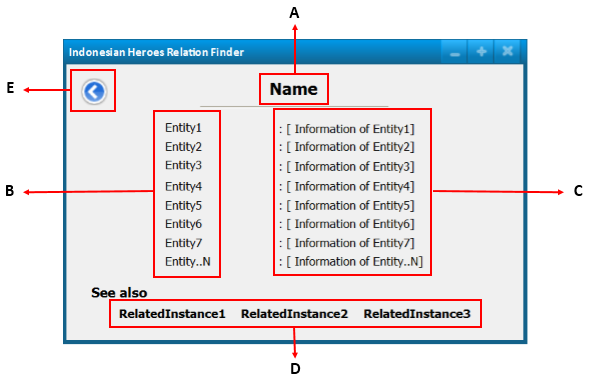
1. Menyatakan istilah dasar yang akan digunakan untuk menentukan ontologi (seperti *class, entity, relation*). Bagian ini sering disebut dengan *meta-ontology* karena pada dasarnya hal itulah yang menjadi dasar ontologi untuk merepresentasikan istilah-istilah utama dalam mengekspresikan ontologi.
2. Memilih bahasa yang mendukung *meta-ontology*.
3. Menulis *code*.

* **Ontology Mapping**

Proses pengintegrasian beberapa ontologi yang sudah ada. Hal yang perlu dilakukan adalah mengidentifikasi kesamaan antara ontologi-ontologi yang memiliki kemiripan domain. Untuk melakukan pengintegrasian atau *ontology mapping*, hal-hal yang dapat dilakukan adalah:

1. Menggabungkan beberapa ontologi menjadi satu ontologi yang koheren.
2. Menyetarakan ontologi-ontologi tersebut dengan cara menentukan hubungan antar elemennya dan mengaitkan informasi yang dimiliki oleh satu elemen dengan elemen yang lain.
3. Mencocokkan elemen yang ada dalam ontologi-ontologi tersebut.

Setelah tahapan-tahapan tersebut diselesaikan, hasil yang diharapkan dari tugas akhir ini adalah terbentuknya *rule* untuk mengetahui jenis hubungan antar tokoh bersejarah yang diimplementasikan dalam suatu aplikasi sederhana. Aplikasi yang dibangun hanya dapat digunakan untuk menampilkan hasil pencarian relasi. Rancangan antarmuka aplikasi ditunjukkan pada **Gambar 10**.

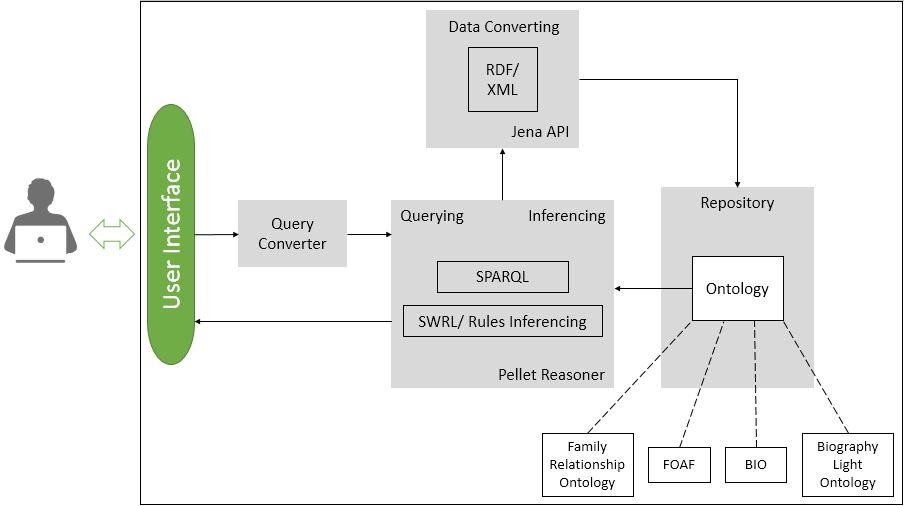


Gambar 10. Rancangan Antarmuka Aplikasi

Keterangan Gambar:

1. Nama tokoh/ i*nstance*/ individu yang dipilih.
2. Entitas yang digunakan oleh *instance* untuk menyatakan ontologi.
3. Informasi dari entitas yang dimiliki *instance*.
4. Hasil pencarian *instance* terkait nama tokoh yang dipilih.
5. Untuk kembali menuju home (list nama tokoh bersejarah).

Aplikasi tersebut tidak dapat digunakan untuk mengelola data karena pengelolaan data dilakukan pada proses *ontology coding* menggunakan *tools* Protege. Adapun arsitektur sistem dari aplikasi yang akan dibangun dapat dilihat pada **Gambar 11**.



Gambar 11. Arsitektur Sistem

# METODOLOGI

## Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

## Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai informasi yang melekat pada tokoh bersejarah, ontologi, OWL, *Family Relationships Ontology*, FOAF *(Friend of a Friend)*, BIO, Biography Light Ontology, SPARQL, Jena API, dan Pellet *reasoner*.

## Analisis dan desain perangkat lunak

Pada tahap analisis akan dilakukan *ontology capture*, yaitu penentuan *classes, properties,* dan *constraints* yang akan dibuat. Setelah analisis selesai, akan dilakukan desain pada tahap *ontology coding* untuk menuliskan ontologi dalam bahasa kode formal yang dimengerti oleh sistem.

## Implementasi perangkat lunak

Dalam tahap ini, dilakukan implementasi berdasarkan rancangan yang dibuat dalam tahap sebelumnya. Aplikasi sederhana untuk menampikan hasil pencarian relasi akan dibangun dengan bahasa Java menggunakan *tools* Microsoft Visual Studio. Sedangkan proses *ontology mapping* dilakukan dengan *tools* Protege 4.3 yang mendukung ekstensi *Web Ontology Language* (OWL).

## Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba aplikasi untuk mencari relasi keterkaitan antar tokoh serta mengadakan perbaikan jika ada kekurangan. Pengujian aplikasi akan dilakukan dengan menggunakan Pellet *reasoner*. Evaluasi dilakukan untuk mengetahui karakteristik dan kecenderungan jalannya sebuah program atas sebuah rangkaian *test case* yang diberikan.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | Tahun | | | | | | | | | | | | | | | | | | | | |
| Desember | | | | Januari | | | | Februari | | | | Maret | | | | | April | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan Evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan Buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | "Ontologies," [Online]. Available: http://www.w3.org/standards/semanticweb/ontology. [Accessed December 2015 ]. |
| [2] | "Ontology (information science)," [Online]. Available: https://en.wikipedia.org/wiki/Ontology\_(information\_science). [Accessed December 2015]. |
| [3] | "OWL," [Online]. Available: http://www.w3.org/2001/sw/wiki/OWL. [Accessed December 2015]. |
| [4] | "National Hero of Indonesia," [Online]. Available: https://en.wikipedia.org/wiki/National\_Hero\_of\_Indonesia. [Accessed December 2015]. |
| [5] | E. Hyvönen, Publishing and Using Cultural Heritage Linked Data on the Semantic Web (Synthesis Lectures on Semantic Web, Theory and Technology), Finland: Morgan & Claypool Publishers, 2012. |
| [6] | M. A. Ramadhanie, "Penerapan Ontologi Objek Pembelajaran Untuk Kebutuhan Personalisasi E-Learning Berbasis Semantic Web," Universitan Indonesia, 2009. |
| [7] | S. J. Miller, "Introduction to Ontology Concepts and Terminology," University of Wisconsin-Milwaukee, Lisbon, 2013. |
| [8] | M. Horridge, "A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.2," The University Of Manchester, Manchester, 2009. |
| [9] | S. Nikles, "Expressiveness of Enterprise Modelling," University of Applied Sciences and Arts Northwester, Switzerland, 2010. |
| [10] | "FOAF (ontology)," [Online]. Available: https://en.wikipedia.org/wiki/FOAF\_(ontology). [Accessed December 2015]. |
| [11] | L. M. Dan Brickley, "FOAF Vocabulary Specification 0.98," 2010. |
| [12] | A. T. A. A.-A. Mumtaz M. Ali Al-Mukhtar, "The Implementation of FOAF Ontology for an Academic Social Network," *International Journal of Computer Science Engineering and Technology( IJCSET),* vol. 4, no. 1, pp. 10-14, 2014. |
| [13] | A. H. Dan Wu, "A method of identifying ontology domain," *Procedia Computer Science,* vol. 35, pp. 504-513, 2014. |
| [14] | E. S. Bijan Parsia, "Pellet: An OWL DL Reasoner," *Software Engineering and the Semantic Web,* vol. 5, no. 2, pp. 51-53, 2007. |
| [15] | "BIO: A vocabulary for biographical information," [Online]. Available: http://vocab.org/bio/. [Accessed December 2015]. |
| [16] | M. R. Ramos, "Biography Light Ontology: An Open Vocabulary For Encoding Biographic Texts," 2009. |
| [17] | M. G. Mike Uschold, "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review,* vol. 11, no. 2, 1996. |
| [18] | C. D. S. G. T. Georgios Paliouras, Knowledge-Driven Multimedia Information Extraction and Ontology Evolution: Bridging the Semantic Gap, 2011: Springer, London. |