Berdasarkan hal tersebut, penulis mencoba untuk meneliti lebih jauh mengenai bagaimana merancang sebuah sistem manajemen pengetahuan gamelan Bali berbasis teknologi web semantik. Teknologi semantik emungkinkan menggambarkan objek dan repositori dalam bentuk ontologi. Ontologi merepresentasikan pengetahuan pada level semantik karena ontologi berisikan entitas semantik (*concept, relation* dan *instance*) sebagai pengganti kata. Selain itu ontologi memungkinkan untuk menspesifikasikan hubungan semantik antar entitas dan juga untuk menyimpan fakta dan aksioma tentang domain pengetahuan (Salton, 1993).

Sistem pencarian pada manajemen pengetahuan gamelan Bali berbasis web semantik yang akan dikembangkan menyediakan fungsi pencarian pengetahuan mengenai gamelan Bali berdasarkan informasi yang ada pada gamelan seperti nama instrumen gamelan, jenis instrumen gamelan, cara memainkan instrumen gamelan, dan informasi lain yang berhubungan dengan gamelan Bali.

Dalam sebuah sistem berbasis web, semantik ontologi digunakan sebagai basis pengetahuan atau metadata yang menerapkan konsep semantik.

* 1. ***Semantic Web***

*Semantic web* merupakan perluasan dari web saat ini, dimana informasi memiliki arti yang terdefinisi secara lebih baik dengan mengupayakan persamaan persepsi antara konsep-konsep yang ada, sehingga memungkinkan manusia dan komputer untuk bekerjasama secara lebih optimal (Berners-Lee., 2001).

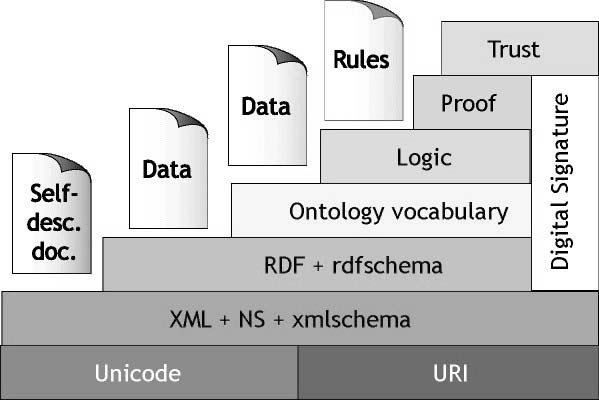
*W3C (World Wide Web Consortium*) memberikan suatu visi dari semantic web yaitu gagasan untuk memiliki data di web yang didefinisikan serta dihubungkan sedemikian rupa sehingga bisa digunakan oleh mesin, bukan hanya untuk ditampilkan tetapi juga untuk tujuan automasi, integrasi dan penggunaan kembali data antar berbagai aplikasi (*W3C*, 2001).

*Semantic web* mengindikasikan bahwa makna data pada web dapat dipahami, baik oleh manusia maupun oleh computer (Passin, 2004). Agar dapat diproses oleh mesin, dokumen web dianotasikan dengan metadata.

* 1. **Arsitektur *Semantic Web***

Menurut *World Wide Web Consortium (W3C)*, arsitektur dari *semantic web* terdiri dari beberapa Layer yang ditunjukkan oleh Gambar 1.

1. *Layer Unicode* dan *URI. Uniform Resource Identifiers (URI)* memastikan penggunaan sekumpulan karakter yang telah disepakati secara internasional dan menyediakan alat untuk mengidentifikasi obyek di semantik web. Jenis *URI* yang terkenal adalah *URL (Uniform Resource Locator)* yang akan memberitahu komputer dimana letak suatu *resource*.
2. *Layer XML, Namespace, XML Schema.* Layer ini mengintegrasikan definisi *Semantic web* dengan dokumen *XML (Extensible Markup Language)* lain yang sesuai standar. *XML* merupakan format standar untuk dokumen terstruktur dan sebagai cara paling fleksibel untuk menciptakan standar bagi format informasi dan kemudian menyediakan format tersebut beserta datanya di web. *XML Schema* menggambarkan struktur dan batasan dari isi dokumen *XML, namespace* merupakan standar yang digunakan untuk menentukan label unik kepada sekumpulan nama elemen yang didefinisikan oleh *XML Schema*.
3. *RDF* dan *RDF Schema. RDF (Resource Description Framework)* dan *RDF Schema* memungkinkan pengguna untuk membuat pernyataan tentang obyek dan *URI* serta mendefinisikan kosakata yang bisa diacu dengan *URI* tersebut. Layer inilah yang menentukan tipe dari *resource* dan *link*. *RDF Schema* mendeklarasikan keberadaan kelas dan properti, termasuk subkelas, subproperti, domain dan range.
4. *Ontology vocabulary. Ontology* mendukung perkembangan kosakata karena *pada Layer RDF* dapat ditentukan relasi antar konsep yang berbeda.
5. *Logic. Layer logic* menyediakan *framework* untuk menulis aksioma dari aturan dasar sistem. Layer ini digunakan untuk meningkatkan bahasa ontologi dan memungkinkan penulisan aplikasi pengetahuan deklaratif khusus.
6. *Proof. Layer proof* mengeksekusi aturan dari *Layer logic*. *Layer proof* melibatkan proses deduktif serta representasi proof dalam bahasa web dan validasi proof.
7. *Trust. Layer trust* mengevaluasi apakah hasil dari Layer proof bisa dipercaya. Layer ini akan muncul melalui penggunaan *digital signature* berdasarkan rekomendasi yang diberikan oleh agen yang terpercaya.
8. *Digital Signature. Digital signature* mendefenisikan blok dari data yang terenkripsi yang akan dimanfaatkan oleh komputer dan agen untuk memastikan apakah suatu informasi yang disediakan oleh sumber yang terpercaya serta mendeteksi adanya perubahan pada dokumen.



**Gambar 1. Arsitektur *Semantic Web***

**(Berners-Lee, 2001)**

* 1. ***Ontology***

*Semantic web* memanfaatan ontologi untuk merepresentasikan basis pengetahuan dan sumber daya *web*. Ontologi menghubungkan simbol-simbol yang dipahami manusia dengan bentuknya yang dapat diproses oleh mesin, dengan demikian ontologi menjadi jembatan antara manusia dan mesin (Davies dkk., 2006).

Ontologi bermanfaat untuk meningkatkan akurasi dalam proses pencarian informasi di web. Mesin pencari dapat mencari halaman yang merujuk pada konsep yang tepat dalam sebuah ontologi. Mesin pencari dapat menemukan dokumen yang relevan, mesin pencari dapat menyarankan pengguna untuk memberikan *query* yang lebih umum (Nurkhamid, 2009). Jika terlalu banyak dokumen yang ditemukan, mesin pencari dapat menyarankan *quer*y yang lebih spesifik (Antoniou & van Harmelen, 2008).

* + 1. **Konsep Ontologi**

Antoniou dan Van Harmelen (2008), mengemukakan bahwa sebuah ontologi didefinisikan sebagai sebuah spefisikasi formal dan eksplisit dari sebuah konseptual. Makna konseptual merujuk pada model abstrak dari sesuatu hal. Eksplisit mengindikasikan bahwa elemen-elemen konseptual harus didefinisikan dengan jelas, dan formal berarti bahwa spesifikasi tersebut harus dapat diproses oleh mesin. Dalam pandangan Gruber ontologi merupakan representasi pengetahuan dari sebuah domain, dengan sekumpulan objek dan relasi dideskripsikan oleh sebuah *vocabulary*.

*W3C* menyebutkan bahwa ontologi adalah sebuah istilah yang diambil dari ilmu filsafat yang merujuk pada bidang ilmu yang mendeskripsikan berbagai entitas dalam dunia dan bagaimana entitas-entitas tersebut saling berelasi (McGuinners & van Harmelen, 2004). Ontologi menyediakan deskripsi untuk elemen kelas-kelas (*classes*) dalam berbagai domain, relasi (*relations*) antar kelas-kelas, dan properti (*property*) yang dimiliki oleh kelas-kelas tersebut.

Ontologi digunakan untuk bidang kecerdasan buatan, representasi pengetahuan, pemrosesan bahasa alami, web semantik, rekayasa perangkat lunak, dan banyak bidang lainya. Dalam sistem informasi, ontologi adalah spesifikasi yang jelas tentang serangkaian konsep yang menjelaskan sebuah wilayah pengetahuan tertentu yang dipakai bersama oleh para pengguna sistem yang bersangkutan.

* + 1. **Manfaat Ontologi**

Beberapa manfaat menggunakan ontologi (Antoniou dan Van Harmelen, 2008), yaitu: 1) Ontologi dapat membagi pemahaman atau definisi tentang konsep-konsep dalam sebuah *domain* (*sharing* informasi); 2) Ontologi menyediakan cara untuk menggunakan kembali domain pengetahuan (*knowledge domain reusable*); 3) Ontologi membuat asumsi eksplisit sebuah *domain*; 4) Ontologi bersama dengan bahasa deskripsi (seperti *RDF Schema*), menyediakan cara untuk mengkodekan pengetahuan dan semantik seperti *machine- understand*; 5) Ontologi memungkinkan pemrosesan mesin otomatis dalam skala besar.

* + 1. ***OWL (Ontology Web Languag*e)**

*OWL (Ontology Web Language)* merupakan suatu bahasa ontologi yang digunakan untuk mendeskripsikan kelas-kelas, properti-properti dan relasi antar objek-objek dalam suatu cara yang dapat diinterpretasi oleh mesin (Breitman dkk., 2007).

*OWL* merupakan sebuah *vocabulary,* namun dengan tingkatan semantik yang lebih tinggi dibandingkan dengan *RDF* dan *RDF Schema*. *OWL* menyediakan tiga sub bahasa yang berbeda tingkatan bahasanya yang dirancang untuk berbagai kebutuhan tertentu dari pengguna, antara lain (Breitman dkk., 2007):

1. *OWL Lite: OWL Lite* menyediakan pendefinisian hirarki kelas dan properti dengan batasan-batasan (*constraints*) yang sederhana. Jenis ini digunakan jika pengguna hanya membutuhkan hirarkis kelas yang sederhana dengan batasan yang sederhana pula.
2. *OWL DL (Description Logic): OWL DL* mendukung pengguna yang menginginkan ekspresi maksimum tanpa kehilangan perhitungan yang lengkap dan ketapatan, *OWL DL* meliputi semua bahasa konstruksi dalam *OWL* dengan batasan tertentu. *OWL DL* dapat menghasilkan hirarkis klasifikasi secara otomatis dan mampu mengecek konsisten dalam suatu ontologi karena *OWL DL* mendukung reasoning.
3. *OWL Full; OWL Full* berguna untuk pengguna yang menginginkan ekspresi maksimum dan kebebasan sintaksis dari *RDF* tanpa ada jaminan perhitungan. *OWL Full* memperbolehkan ontologi untuk meningkatkan arti dari kosakata yang belum digambarkan (*RDF* atau *OWL*). *OWL Full* diperuntukkan bagi user yang menginginkan sub bahasa yang sangat ekspresif dan secara sintaks lepas dari RDF tanpa jaminan komputasional.
   * 1. **Struktur Dokumen *OWL***
4. *Namespace.* Sebelum kita dapat menggunakan kumpulan istilah (*term*) dalam ontologi, kita perlu terlebih dahulu mendefinisikan secara tepat kosakata apa yang akan digunakan. Komponena awal dari standar ontologi adalah mendeklarasikan *namespace*. *Namespace* menyediakan sarana untuk pengidentifikasi dan membuat ontologi lebih mudah dibaca. Berikut ini contoh cuplikan *script OWL* untuk mendeklarasikan sebuah *namespace*

<rdf:RDF

xmlns

[="http://www.w3.org/TR/2004/REC](http://www.w3.org/TR/2004/REC-)- owl-guide-20040210/wine#"

xmlns:vin

[="http://www.w3.org/TR/2004/REC](http://www.w3.org/TR/2004/REC-)- owl-guide-20040210/wine#"

xml:base

[="http://www.w3.org/TR/2004/REC](http://www.w3.org/TR/2004/REC-)- owl-guide-20040210/wine#"

[xmlns:food="http://ww](http://www.w3.org/TR)w.w3.org/TR

/2004/REC-owl-guide- 20040210/food#"

xmlns:owl

[="http://www.w3.org/2002/07/owl#](http://www.w3.org/2002/07/owl) "

xmlns:rdf

[="http://www.w3.org/1999/02/22](http://www.w3.org/1999/02/22-)- rdf-syntax-ns#"

[xmlns:rdfs="http://www.w3.org/20](http://www.w3.org/20) 00/01/rdf-schema#"

xmlns:xsd

[="http://www.w3.org/2001/XMLSchema#">](http://www.w3.org/2001/XMLSchema)

1. *Header.* Dokumen *OWL* biasanya disebut juga *OWL ontology*, memiliki elemen *root* berupa *tag owl: Ontology*. Berikut ini adalah contoh cuplikan *script header* dalam dokumen *OWL*:

<owl:Ontology rdf:about="">

<rdfs:comment>An example OWL ontology</rdfs:comment>

<owl:priorVersion [rdf:resource="http://www.w3.org/TR/20](http://www.w3.org/TR/20) 03/PR-owl-guide-20031215/wine"/>

<owl:imports [rdf:resource="http://www.w3.org/TR/20](http://www.w3.org/TR/20) 04/REC-owl-guide-20040210/food"/>

<rdfs:label>Wine Ontology</rdfs:label>

...

1. Elemen *Class*. *OWL* menyediakan konsep untuk mendefinisikan kelas beserta dengan batasan-batasan (*constraints*) dan aksiomanya (*axiom*). Kelas-kelas (*classes*) dalam dokumen *OWL* didefinisikan dengan menggunakan *owl:Class*. Setiap kelas dalam *OWL* merupakan subkelas dari *owl:Thing*. Berikut ini adalah cuplikan dari *script OWL* untuk mendefinisikan sebuah kelas dan subkelas dalam ontologi:

<owl:Class rdf:ID="Person"/>

<owl:Class rdf:ID="Publication"/>

<owl:Class rdf:ID="Person">

<rdfs:subClassOf rdf:resource="#Thing" />

...

</owl:Class>

Bechhofer, dkk (2004) mengatakan kelas *OWL* pada prinsipnya sama dengan kelas yang didefinisikan di *RDF*. *OWL* menyedikan mekanisme untuk mengklasifikasikan sejumlah *resource* dengan karakteristik tertentu. Setiap kelas *OWL* yang didefinisikan dapat berhubungan dengan sejumlah individu yang disebut perluasan kelas (*class extension*).

* + 1. ***RDF***

*Resource Description Framework (RDF)* merupakan sebuah model data yang sederhana dan fleksibel untuk mendeskripsikan hubungan antara sumberdaya-sumberdaya web dalam bentuk *RDF statement* (Breitman dkk., 2007). *RDF* mendukung interoperabilitas antar aplikasi yang melakukan pertukaran informasi dan bersifat machine-understandable di web. *RDF* menggunakan graf untuk merepresentasikan kumpulan pernyataan. Simpul dalam graf mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas. RDF didasarkan pada gagasan dimana hal-hal yang sedang diuraikan memiliki properti yang didalamnya mempunyai nilai-nilai dan *resource* yang dapat diuraikan dengan pembuatan statement (Manola dan Miller, 2004).

*RDF* menggunakan istilah tertentu untuk menguraikan suatu statement. Bagian yang mengidentifikasi dalam statement dapat disebut subject, karakteristik (*property*) dari *subject* disebut sebagai predicate, sedangkan nilai dari *property* disebut sebagai *object*.

Lassila dan Swick (1999) menyatakan model data *RDF* terdiri atas tiga objek tipe: 1) *Resource*, segala sesuatu yang digambarkan dengan *RDF* disebut resource. Resource bisa berupa keseluruhan atau bagian dari sebuah halaman web. *Resource* ini biasanya diberi nama menggunakan *URI (Uniform Resource Identifier)*. *URI* bersifat bisa diperluas maka *URI* bisa digunakan sebagai pengenal bagi berbagai macam entitas; 2) Properti (*Property*), properti merupakan aspek atau karakteristik, atribut, serta relasi khusus yang digunakan untuk menggambarkan sebuah *resource*. Setiap properti memiliki arti khusus, mendefinisikan nilai yang mungkin, tipe *resource* yang digambarkan dan relasinya dengan properti lain. Pernyataan (*Statement*), suatu *resource* bersama dengan properti dan nilai dari suatu properti untuk *resource* membentuk suatu pernyataan *RDF*. Ketiga bagian ini disebut subjek, predikat dan obyek, yang membentuk *RDF triple*. Obyek dapat berupa *resource* lain, atau berupa literal (*string* sederhana atau tipe data primitif lain yang didefinisikan oleh *XML*).

**.2 Metode Pengambilan Data**

Metode pengambilan data dimaksudkan untuk mengumpulkan data-data pendukung yang dibutuhkan dalam proses pembuatan sistem manajemen pengetahuan gamelan Bali ini. Beberapa cara yang dapat digunakan adalah melalui studi literatur, yaitu dengan mengumpulkan dan mempelajari sejumlah referensi dari berbagai sumber yang berkaitan dengan judul penelitian yang dilakukan.

**.3 Kerangka Konsep Penelitian**

Sistem manajemen pengetahuan gamelan Bali merupakan sistem yang membantu pengguna dalam melakukan penjelajahan dan pencarian pengetahuan gamelan Bali dengan informasi yang relevan dan lengkap.

Sistem manajemen pengetahuan gamelan Bali ini akan memberikan hasil penjelajahan yang terdiri dari hasil pencarian utama dan hasil pencarian terkait berup encarian atau pencarian terkait yang sesuai dan masih berhubungan deng unci atau judul yang ingin dicari oleh pengguna. Hasil pencarian terkait yang ditampilkan oleh sistem terdiri dari nama instrumen gamelan dan informasi mengenai instrumen gamelan tersebut.

* 1. Pemodelan Ontologi

Pemodelan ontologi adalah pemodelan dengan menggunakan *tool Protégé* *4.3*. Sehingga menghasilkan file *OWL* (*Web Ontology Language*).

* 1. Penentuan *domain*

Domain yang ditentukan pada sistem ini adalah pengetahuan mengenai gamelan Bali.

* 1. Penentuan *class*

Dalam menentukan *class*, dikerjakan oleh Made Wardana (2016) dalam penelitiannya yang berjudul “Implementasi *K-Nearest Neighbour* (*KNN* Klasifikasi Artikel Wikipedia Indonesia”.

. Mendefinisikan *slot* atau *properties*

Mendefinisikan *properties* yang digunakan sebagai penghubung antar *class* atau antar *instanse* atau *individual*.

e. Membuat *instances*

Membuat *instances* atau *individual* untuk tiap-tiap *class*.

Pengkodean

Pada tahap pengkodean, dilakukan proses mengintegrasikan file *OWL* dengan bahasa pemrograman *PHP* dan *SPARQL* untuk bahasa *query*nya. Selain itu juga menggunakan *library EasyRDF* sebagai *parser* dari file *OWL*.

. Pengujian Sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, sistem tersebut diuji terlebih dahulu sebelum digunakan. Pengujian dilakukan dengan 2 (dua) cara, yaitu 1) memastikan apakah hasil dari sistem sesuai dengan yang diinginkan dengan perencanaan sebelumnya, dan 2) pengujian terhadap keakuratan hasil dari sistem.

1. Evaluasi Sistem

Evaluasi dilakukan untuk mengetahui apakah sistem yang telah jadi sesuai dengan yang diharapkan. Selain evaluasi juga dilakukan perbaikan sistem sesuai dengan kebutuhan dan hasil pengujian sistem.

7. Penggunaan Sistem

Perangkat lunak yang telah diuji dan diterima pengguna dan siap untuk digunakan.

**Metode Pengujian**

Pengujian yang dilakukan untuk menguji sistem dilakukan dengan 2 (dua) langkah pengujian, yaitu validasi dan pengujian akurasi sistem. Pengujian bertujuan untuk memastikan bahwa sistem yang dibangun telah berjalan sesuai dengan metode *semantic web* yang digunakan. Berikut ini kedua langkah pengujian tersebut.

1. Validasi

Uji coba validasi dikatakan berhasil apabila fungsi yang ada pada sistem sesuai dengan yang diharapkan pengguna. Validasi sistem merupakan kumpulan seri uji coba *Black Box* yang menunjukkan sesuai dengan yang diperlukan.

1. Pengujian Akurasi Sistem

Dalam pengujian ini, dilakukan untuk mengetahui keakuratan aplikasi dalam menampilkan data hasil pencarian oleh pengguna. Metode yang digunakan adalah *semantic browsing* dan *semantic searching*.