# BAB II
# TINJAUAN PUSTAKA

## A. Semantic Web

Pada tahun 1992, Tim Berners-Lee menciptakan World Wide Web Consortium (W3C) dengan tujuan untuk mengembangkan dan menstandarisasi *web*. Penelitian yang dilakukan oleh W3C pada akhirnya mengarah pada pengembangan *semantic web* secara konseptual. Menurut Tim Berners-Lee, James Hendler, dan Ora Lassila [9], *semantic web* merupakan *extension* dari *web* yang ada sekarang ini, di mana informasi memiliki makna yang terdefinisi dengan baik sehingga memungkinkan komputer untuk menampilkan hasil pencarian informasi yang lebih relevan dan mengintegrasikan data dari berbagai sumber yang berbeda. Tim Berners-Lee, James Hendler, dan Ora Lassila [9] mengidentifikasikan tiga komponen penting agar *semantic web* berfungsi dengan baik, yaitu.

a. *Knowledge Representation*

*Knowledge representation* merupakan sekumpulan informasi yang terstruktur dan *inference rules* yang dapat digunakan untuk melakukan *automated reasoning*.

b. *Ontologies*

Ontologi merupakan sebuah dokumen yang secara formal menjelaskan tentang *class-class* dari objek-objek dan mendefinisikan hubungan di antara *class-class* tersebut.

c. *Agents*

*Agents* merupakan program-program yang dapat mengumpulkan *content* dari berbagai sumber yang berbeda dan saling bertukar hasil dengan program-program yang lain.

Menurut Tshering Dema [8], *semantic web* bertujuan untuk membuat informasi dapat dimengerti oleh mesin sehingga mesin tersebut dapat melakukan kerja yang melibatkan *knowledge representation* dan integrasi data secara otomatis.

Menurut Niko Ibrahim [10], *semantic web* memiliki isi *web* yang tidak hanya dapat diekspresikan di dalam bahasa alami yang dimengerti manusia, tetapi juga di dalam bentuk yang dapat dimengerti, diinterpretasi, dan digunakan oleh *software agents*. Dengan demikian, melalui *semantic web*, berbagai perangkat lunak dapat mencari, membagi, dan mengintegrasikan informasi.

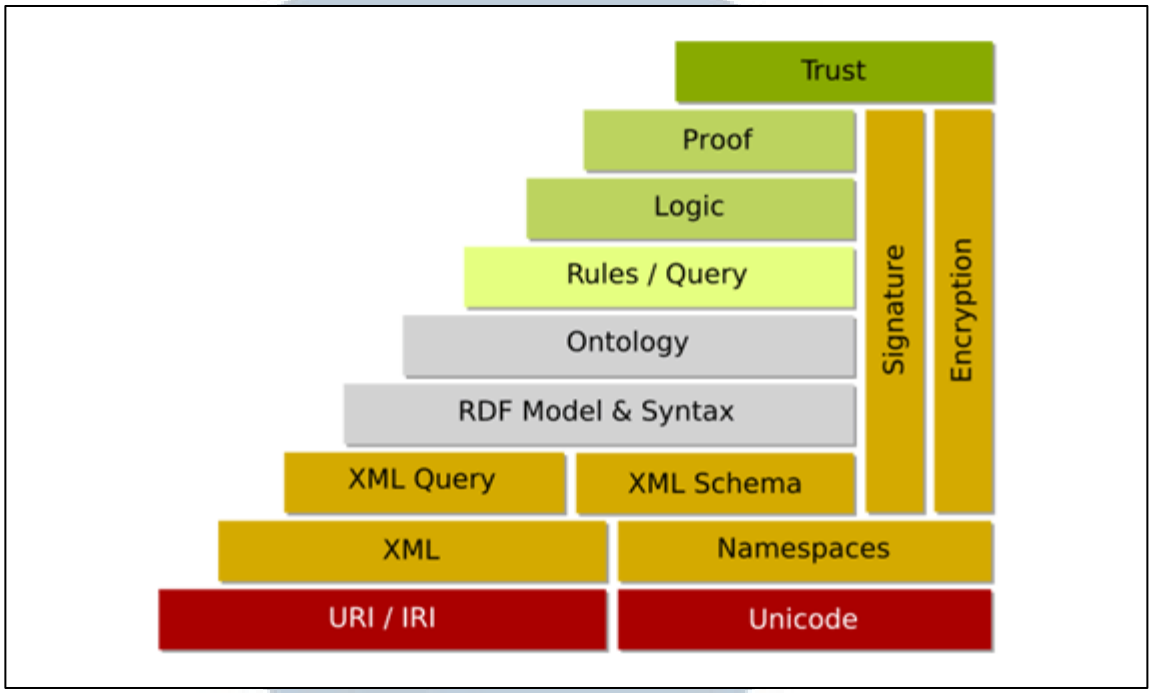
## B. Ontologi

Menurut Brooke Abrahams [6], ontologi terdiri dari istilah-istilah dasar, relasi antara istilah-istilah tersebut, dan aturan-aturan yang mengombinasikan istilah-istilah tersebut. Ontologi tersebut dapat disimpulkan menjadi sebuah *knowledge* yang dapat di-*share* dan digunakan oleh beberapa aplikasi *software*.

Menurut Tshering Dema [8], ontologi merepresentasikan dunia nyata secara terstruktur dan sistematis. Ontologi menyediakan sebuah *reference model* untuk domain-domain ontologi tersebut. *Reference model* merupakan sekumpulan istilah yang dapat digunakan untuk menyederhanakan komunikasi antar domain *experts* dan meningkatkan pemahaman dan *knowledge sharing*.

## C. Komponen-komponen Semantic Web

*World Wide Web Consortium* (W3C) mengembangkan standar-standar yang memungkinkan pembuatan *semantic web* dilakukan. Tim Berners-Lee [11] membuat *semantic web stack* yang merupakan arsitektur dari *semantic web*. *Semantic web stack* menunjukkan hierarki dari komponen-komponen yang membangun *semantic web* dan membuktikan bahwa *semantic web* tidak menggantikan *web* yang telah ada sebelumnya, tetapi merupakan *extension* dari *classical hypertext web*. *Semantic web stack* yang menunjukkan *layer-layer* pada *semantic web* dapat dilihat pada gambar 2.1.




Gambar 2.1 Semantic Web Stack [11]

Menurut Niko Ibrahim [10], komponen-komponen yang paling penting dalam membangun *semantic web* adalah XML, XML Schema, RDF, OWL, dan SPARQL.

a. XML dan XML Schema

Extensible Markup Language (XML) merupakan *general purpose markup language* untuk dokumen-dokumen yang mengandung informasi yang terstruktur. Sebuah dokumen XML mengandung elemen-elemen yang dapat memiliki atribut dan *content*. Di samping itu, elemen-elemen tersebut juga dapat dibuat secara *nested*. John Hebeler dkk. [12] berpendapat bahwa XML merupakan cara yang paling efektif dan banyak digunakan dalam

pertukaran informasi. Menurut Brooke Abrahams [6], XML merupakan *syntax* yang *powerful* dan fleksibel untuk dokumen yang terstruktur, tetapi tidak memiliki *constraint* semantik pada makna dari dokumen-dokumen tersebut.

XML Schema merupakan bahasa yang digunakan untuk mendefinisikan aturan-aturan (*schema-schema*) dari sekumpulan dokumen XML tertentu. Struktur dari dokumen XML yang dibuat harus sesuai dengan *schema* yang telah didefinisikan.

b. RDF dan RDF Schema

Pada *semantic web*, informasi direpresentasikan sebagai sekumpulan pernyataan yang disebut dengan *statements*. Pernyataan-pernyataan tersebut terdiri dari tiga bagian, yaitu subjek, predikat, dan objek sehingga pernyataan-pernyataan tersebut juga sering disebut sebagai *triples*. Subjek dari sebuah *statement* merupakan *resource* yang dideskripsikan oleh *statement* tersebut, sedangkan predikat menjelaskan hubungan antara subjek dan objek. Pernyataan-pernyataan tersebut dapat membentuk sebuah *directed graph*, dengan subjek dan objek dari setiap *statement* sebagai *nodes* dan predikat sebagai *edges*. Inilah *data model* yang digunakan oleh *semantic web* dan ditulis secara formal dalam sebuah bahasa yang disebut dengan Resource Description Framework (RDF).

Menurut Niko Ibrahim [10], mekanisme pendeskripsian *resource* ini merupakan komponen utama yang dikemukakan oleh *W3C's semantic web* di mana perangkat lunak dapat menyimpan, menggunakan, dan saling bertukar informasi yang dapat dimengerti oleh mesin dan didistribusikan melalui *web* sehingga memampukan pengguna dalam menangani informasi tersebut dengan tingkat efisiensi dan tingkat kepastian yang lebih baik.

John Hebeler dkk. [12] mengemukakan bahwa meskipun RDF menyediakan model yang *powerful* dan *expressive* untuk menangkap informasi, RDF sendiri tidak menyediakan cara untuk menangkap makna dari informasi tersebut. RDF Schema (RDFS) menyediakan *vocabulary* yang spesifik untuk RDF yang dapat digunakan untuk mendefinisikan taksonomi dari *class-class*, *properties*, *simple domain*, dan spesifikasi *range* untuk *properties*. RDFS tidak mendefinisikan *shared vocabularies* tersebut, tetapi menyediakan sebuah bahasa yang dapat digunakan untuk mengembangkan *shared vocabularies* tersebut. RDFS *vocabularies* menjelaskan *class-class* dari *resources*, dan *properties* yang digunakan dalam model RDF. Dengan menggunakan RDFS, *semantic web developer* dapat menyusun *classes* dan *properties* dalam hierarki generalisasi atau spesialisasi, mendefinisikan domain dan *range* yang diharapkan untuk *properties*, menyatakan *class membership*, dan menspesifikasikan tipe data.

c. OWL

Web Ontology Language (OWL) merupakan bahasa yang dibuat untuk meng-*extend* RDF Schema yang terbatas untuk mendefinisikan domain aplikasi. *Ontology* sendiri merupakan cara untuk mendeskripsikan makna dan relasi dari suatu istilah. Menurut Lee Provoost dan Erwan Bornier [13], deskripsi tersebut dapat membantu sistem komputer dalam menggunakan istilah-istilah tersebut dengan lebih mudah.

Menurut Deborah L. McGuinness dan Frank van Harmelen [14], OWL menambahkan *vocabulary* untuk mendeskripsikan *properties* dan *classes*, seperti "among others", relasi antar *class* (contoh: disjointness), kardinalitas (contoh: "exactly one"), *equality*, *richer typing of properties*, karakteristik *properties* (contoh: symmetry), dan *enumerated classes*.

Menurut Jeffrey T. Pollock [15], OWL memiliki tiga subbahasa, yaitu OWL-Lite, OWL-DL, dan OWL-Full. OWL-Lite digunakan dalam situasi di mana hanya dibutuhkan *constraints* dan hierarki *class* yang sederhana, sedangkan OWL-DL memungkinkan penggunaan bahasa OWL yang utama secara penuh, tetapi dengan beberapa pembatasan pada *class restrictions*. Di sisi lain, OWL-Full digunakan dalam situasi di mana tingkat ekspresi yang sangat tinggi lebih diutamakan daripada komputasi yang dibutuhkan.

```
<owl:Class rdf:ID="Book"/>

<owl:Class rdf:ID="JohnWileyBook">
 <rdfs:subClassOf rdf:resource="Book"/>
</owl:Class>

<owl:Class rdf:ID="ForDummies">
 <rdfs:subClassOf rdf:resource="JohnWileyBook"/>
</owl:Class>

<owl:Class rdf:ID="HigherEducation">
 <rdfs:subClassOf rdf:resource="JohnWileyBook"/>
</owl:Class>

<owl:Class rdf:ID="InterScience">
 <rdfs:subClassOf rdf:resource="JohnWileyBook"/>
</owl:Class>
```

Gambar 2.2 Contoh Struktur OWL [15]

Gambar 2.2 menunjukkan contoh struktur *class* dan *subclass* pada OWL. Berdasarkan contoh tersebut, *class* Book memiliki *subclass* JohnWileyBook dan *class* JohnWileyBook memiliki *subclass* ForDummies, HigherEducation, dan InterScience.

d. SPARQL

Simple Protocol and RDF Query Language (SPARQL) merupakan sebuah *query language* untuk RDF. SPARQL menyediakan *interface* yang deklaratif untuk berinteraksi dengan *database* RDF. SPARQL merupakan bahasa seperti SQL, tetapi menggunakan RDF *triples* dan *resources* dalam mencocokkan bagian dari *query* dan mengembalikan hasil dari *query*. Karena RDFS dan OWL dibangun berdasarkan RDF, SPARQL dapat digunakan untuk melakukan *query* pada *ontologies* dan *knowledge bases* secara langsung. SPARQL bukan hanya sebuah *query language*, melainkan juga sebuah protokol untuk mengakses data RDF.

```
PREFIX fb:<http://rdf.freebase.com/ns/>

SELECT ?film ?when
WHERE {

   ?film fb:film.film.initial_release_date ?when .
      FILTER (?when > "2002")
}
```

Gambar 2.3 Contoh Penggunaan FILTER pada SPARQL [20]

Gambar 2.3 menunjukkan contoh penggunaan FILTER pada SPARQL. Berdasarkan contoh tersebut, *query* akan mengembalikan *resource* film yang di-*release* setelah tahun 2002.

```
PREFIX fb: <http://rdf.freebase.com/ns/>

SELECT ?film ?reldate
WHERE {
     ?film fb:film.film.directed_by fb:en.ron_shelton .
     OPTIONAL { ?film fb:film.film.initial_release_date ?reldate .}
}
```

Gambar 2.4 Contoh Penggunaan OPTIONAL pada SPARQL [20]

Gambar 2.4 menunjukkan contoh penggunaan OPTIONAL pada SPARQL. Berdasarkan contoh tersebut, jika suatu *resource* film memiliki *release date*, OPTIONAL *clause* akan melakukan *binding* terhadap *release date* ke dalam hasil *query*.

```
PREFIX fb:<http://rdf.freebase.com/ns/>

SELECT ?name
WHERE {

   {
      ?film fb:film.film.directed_by ?director .
      ?director fb:type.object.name ?name
         filter regex(?name, "^... ", "i")
   }
UNION
  {
     ?film fb:film.film.starring ?actor .
     ?actor fb:type.object.name ?name
        filter regex(?name, "^b", "i")
  }
}
```

Gambar 2.5 Contoh Penggunaan UNION pada SPARQL [20]

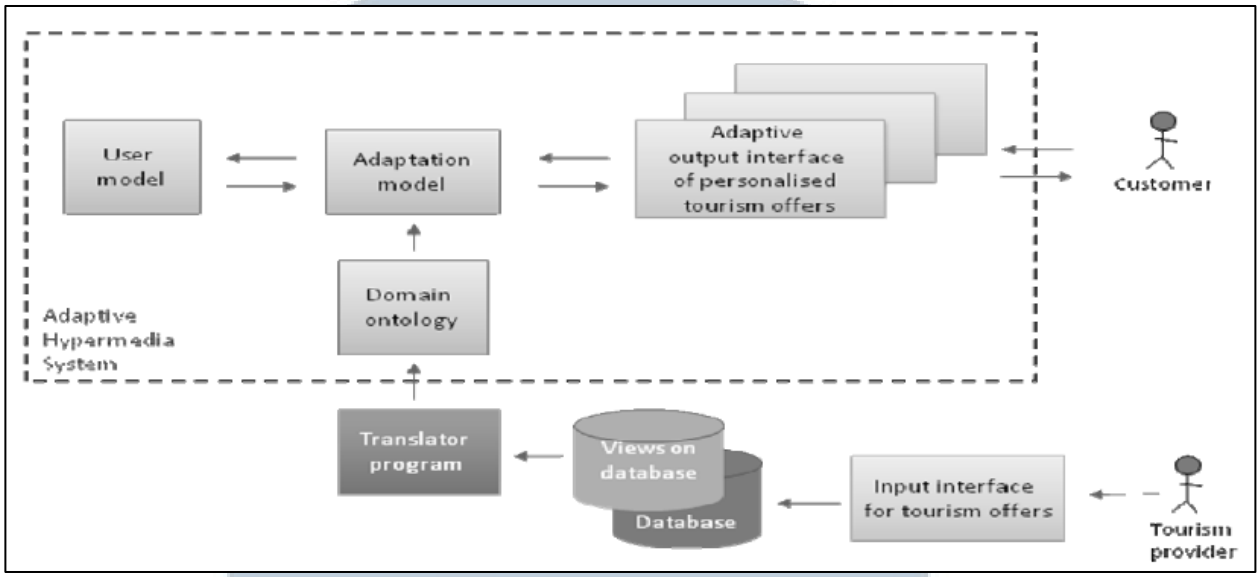
Gambar 2.5 menunjukkan contoh penggunaan UNION pada SPARQL. Berdasarkan contoh tersebut, *query* akan mengembalikan nama *director* yang memiliki *first name* yang terdiri dari tiga huruf dan nama *actor* yang diawali dengan huruf 'b'.

e. RAP-RDF

Menurut Radoslaw Oldakowski, Christian Bizer, dan Daniel Westphal [26], RAP-RDF merupakan *semantic web toolkit* yang menyediakan fitur untuk melakukan *parsing*, *manipulating*, *storing*, *querying*, *serving*, dan *serializing graph* RDF untuk PHP *developer*. RAP dimulai sebagai sebuah *open source project* oleh Freie Universität Berlin pada tahun 2002 dan telah dikembangkan oleh *semantic web community*. Versi terakhir dari RAP-RDF API adalah versi 0.9.6. RAP-RDF dibangun dan dikembangkan karena *semantic web* memungkinkan terjadinya *sharing* informasi secara global dan memungkinkan *user* untuk membuat dan melakukan *sharing knowledge*.

## D. Model Semantic Web untuk E-Tourism

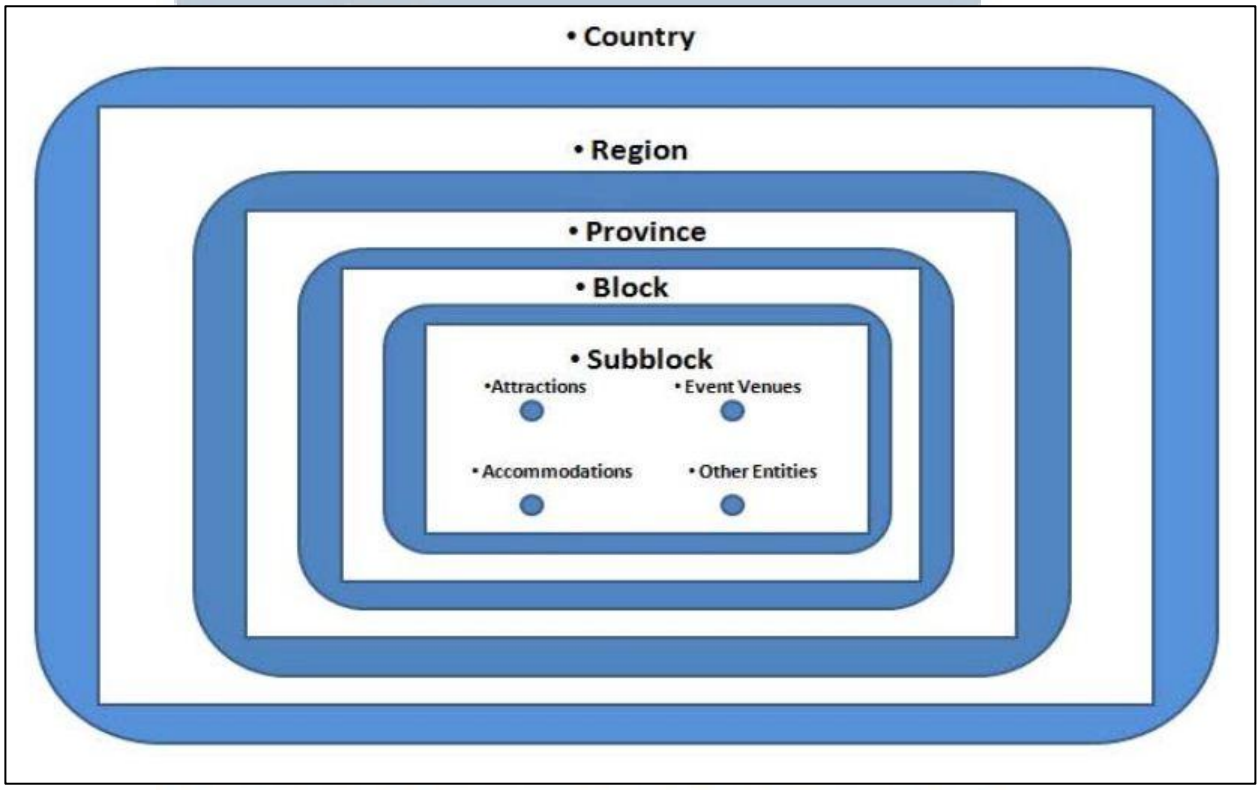
*Tourism provider* melakukan *input* data-data pariwisata ke dalam *database* melalui suatu *input interface*. Setelah itu, data-data yang terdapat di dalam *database* tersebut digunakan untuk membuat *ontology* dengan menggunakan suatu *translator program*. *Ontology* tersebut akan digunakan untuk memodelkan domain *knowledge*. Dalam sistem yang adaptif, sebuah *user model* untuk merepresentasikan *users* (*customers*) dan sebuah *adaptation model* didefinisikan. *Adaptation model* tersebut menggunakan *knowledge* dari *ontology* untuk menghasilkan *output* yang dipersonalisasi dan disesuaikan berdasarkan *user model* yang diinisialisasi mengikuti *customers behavior* terhadap *interface*. *Output* yang dihasilkan berupa *suggestions* mengenai informasi pariwisata yang sesuai dengan keinginan *customers*. *Output* inilah yang dijadikan parameter sebagai acuan untuk melakukan uji coba. Alur proses *semantic web* ini dapat dilihat pada gambar 2.6.




Gambar 2.6 Model *Semantic Web* untuk *E-Tourism* [17]

## E. Model Data untuk E-Tourism

Menurut Tshering Dema [8], data-data akomodasi, objek wisata, dan *event venues* dikelompokkan berdasarkan daerah (*city*, *town*, *village*, dan *place*) yang terdapat di suatu *block*. *Block* merupakan wilayah yang lebih kecil daripada provinsi dan provinsi merupakan bagian dari *region* yang terdapat di Negara Bhutan. Model data yang digunakan dapat dilihat pada gambar 2.7.




Gambar 2.7 Model Data eTourPlan [8]

## F. E-Tourism

Menurut Theophilus Wellem [3], teknologi informasi telah menjadi salah satu komponen penting dalam berbagai bidang dan industri, termasuk pariwisata. *E-tourism* merupakan pariwisata berbasis teknologi informasi (*IT-enabled tourism*). Menurut Riina Henriksson [23], pariwisata merupakan industri yang melibatkan informasi yang sangat banyak yang dibutuhkan dalam suatu perjalanan wisata. Informasi-informasi tersebut antara lain akomodasi, *attraction* / objek wisata, dan *event*. Menurut Kristine Corcoran dkk. [24], akomodasi, *tourist attractions*, dan *events* merupakan elemen-elemen pariwisata yang meningkatkan aktivitas ekonomi dalam suatu wilayah.

## G. Dasar Teori Pendukung Variabel Penelitian

Menurut Kathrin Prantner, Katharina Siorpaes, dan Daniel Bachlechner [16], *semantic web search* yang dibuat berhasil memberikan *suggestions* yang sesuai dengan keinginan para wisatawan sehingga para wisatawan tersebut tidak membuang banyak waktu dalam mencari informasi yang dibutuhkan.

Magnus Niemann, Malgorzata Mochol, dan Robert Tolksdorf [7] berhasil mengimplementasikan teknologi *semantic web* untuk mengoptimalkan proses pencarian hotel. Proses pencarian tersebut dapat menampilkan hasil pencarian berupa hotel-hotel yang sesuai dengan kebutuhan *customer* berdasarkan

variabel-variabel, seperti lokasi hotel, biaya, dan fasilitas-fasilitas yang dimiliki oleh hotel-hotel tersebut.

Aplikasi eTourPlan yang berhasil dibuat oleh Tshering Dema [8] memiliki *search engine* yang dapat membantu para wisatawan dalam melakukan pencarian informasi mengenai akomodasi, objek wisata, dan *event-event* yang terdapat di Bhutan.