

BAB III

ANALISA PERANCANGAN ONTOLOGI

3.1 Analisis Sistem

Berbagai hal dapat menyebabkan suatu penyakit sulit terasi, dari faktor lingkungan hingga kurang meratanya tenaga medis karena sulitnya akses menuju daerah-daerah. Hal tersebut membuat terlambatnya penanganan dan dapat timbul peularan. Sehingga perlu adanya teknologi yang memberikan informasi mengenai gejala – gejala disertai penyakit yang ditimbulkan

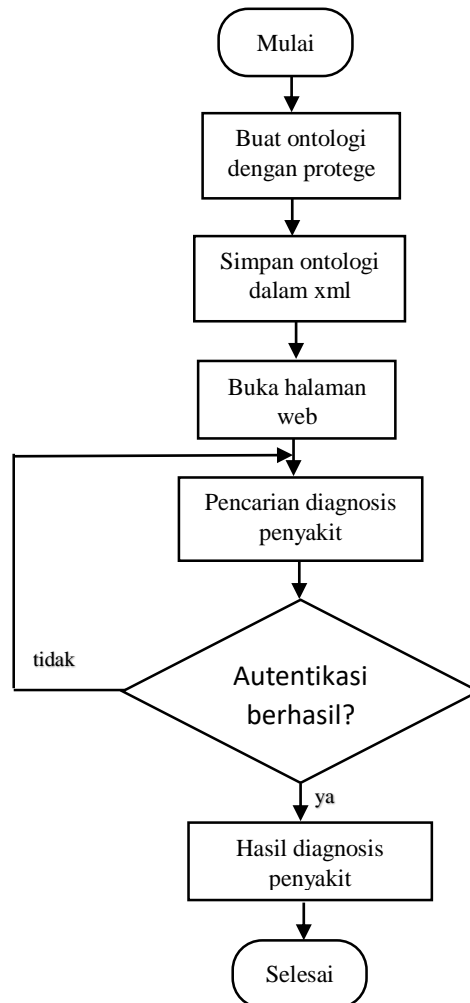
Penerapan aplikasi diagnosis penyakit menggunakan semantik ontologi dapat menghasilkan informasi yang banyak mengenai gejala penyakit yang memudahkan masyarakat didaerah-daerah dapat mengenali suatu penyakit sehingga tidak ada keterlambatan dalam penanganannya. Data penyakit, gejala, dan penularannya dapat diperoleh berdasarkan buku yang berhubungan dengan penyakit yang didinginkan, perawat, dan dokter. Sehingga data yang diberikan akurat.

3.2 Hasil Analisis

Hasil analisis dari pencarian keterkaitan gejala – gejala yang dirasakan untuk mengetahui penyakit yang diderita dengan variabel yang digunakan yaitu : gejala, penyakit, dan penularan. Dengan adanya pencarian ini diharapkan dapat mengali pola-pola yang dapat dijadikan informasi baru atau rekayasa pengetahuan baru untuk sentuhan teknologi di daerah yang masih minim tenaga medis. Semantik *ontology* digunakan karena menurut Mutawalli (2018), merancang sistem untuk pendeteksian penyakit infeksi tropis berbasis ontologi penelitian tersebut bertujuan untuk mengendalikan penyakit menular khususnya dengan kategori infeksi tropis. Nurul Rohmawati W, Sofi Defianti dan Mohammad Zazuli, 2015 dalam penelitian tentang “Pemodelan dan Pengembangan Sistem Pendeteksian Penyakit Infeksi Tropis Berbasis Ontologi” dapat mendeteksi penyakit demam tifoid, demam berdarah, dan malaria dengan mendeteksi gejala dan cara penanganannya.

3.2.1 Deskripsi Sistem

Diagram alur sistem menunjukkan bagaimana sistem melakukan proses eksekusi dimana pengguna memberikan masukan dan sistem merespon masukan tersebut. Diagram alur sistem ditunjukkan pada Gambar 3.1.

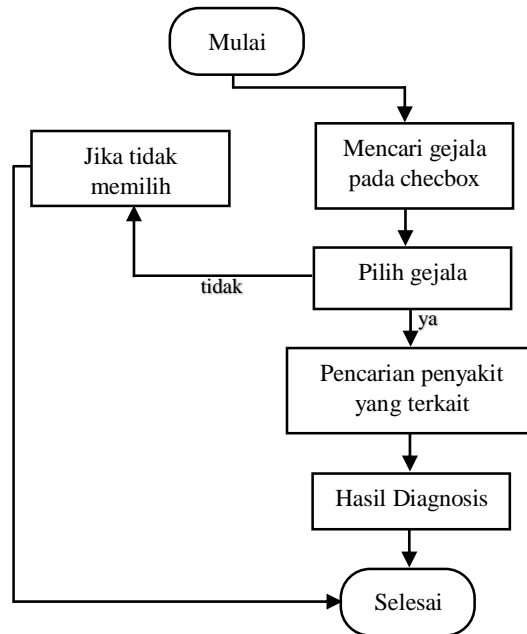


Gambar 3.1 Diagram alur domain

Pada Gambar 3.1 merupakan diagram alur dari *domain expert* atau pengguna ke aplikasi bagaimana membuat aplikasi diagnosis penyakit.

Proses pencarian diagnosis penyakit pada aplikasi rekayasa pengetahuan ini dilakukan agar sistem memahami yang dimaksudkan

oleh pengguna. Adapun alur pencarian diagnosis penyakit ditunjukkan pada gambar 3.2.



Gambar 3.2 Alur system pencarian diagnosis penyakit

Penjelasan Gambar 3.2 :

1. Mencari gejala yang dirasakan.
2. Memilih gejala dengan memberi tanda centang pada *checkbox* yang disediakan.
3. Inisialisasi data gejala dengan semantik ontologi yakni menampilkan informasi penyakit yang berkaitan dengan gejala yang telah dipilih disertai dengan penularannya.

3.2.2 Kebutuhan Data

Menyiapkan data penyakit yang diinginkan beserta gejala dan penularan. Literature diambil dari buku maupun wawancara dengan ahlinya, baik perawat maupu dokter.

3.3 Pemodelan Sistem

Pemoedelan sistem dilakukan agar ontologi dalam diagnosis penyakit yang dibuat sesuai yang diharapkan.

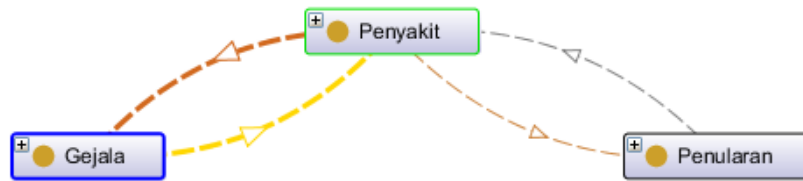
3.3.1 Proses Semantik Ontologi

Proses semantik ontologi akan menggunakan data pada tabel 3.2. berikut langkah – langkah yang perlu dilakukan dalam menerapkan semantik ontologi (Ayuningtyas , 2009).

1. Menentukan domain dalam batasan ontologi. Dimana domain akan melingkupi ontologi dan mengapa ontologi digunakan.
2. Mempertimbangkan penggunaan ontologi yang sudah ada.
3. Menentukan istilah yang akan dijelaskan dan mudah dipahami *user*.
4. Mendefinisikan *class* ontologi dan menyusun *class* dalam hirarki taksonomi (*subclass – superclass*). Terdapat beberapa pendekatan dalam pengembangan hirarki *class* (Matthew et al., 2004) yaitu : Proses pengembangan *top-down, bottom-up, combination*. Dalam penelitian ini menggunakan pengembangan *top-down*.
5. Mendefinisakan *slot* atau *properties* dan menjabarkan nilai dari *slot* tersebut.
6. Mendefinisikan *facets* pada *slots*. Untuk melengkapi *properties*.
7. Membuat *instances* atau individual. Dalam mendefinisakan individual *instance* pada *class* harus diperhatikan persyaratan berikut:
 1. Pemilihan *class*
 2. Pembuatan individual *instances* pada *class* tersebut.
Knowledge base selanjutnya dapat dibuat dengan mendefinisikan individual *instances* dari *class* yang terisi pada nilai spesifik *slot* dan *slot* batasan tambahan (*facets*).
 3. Mengisi nilai *slot*.

3.4 Perancangan Sistem

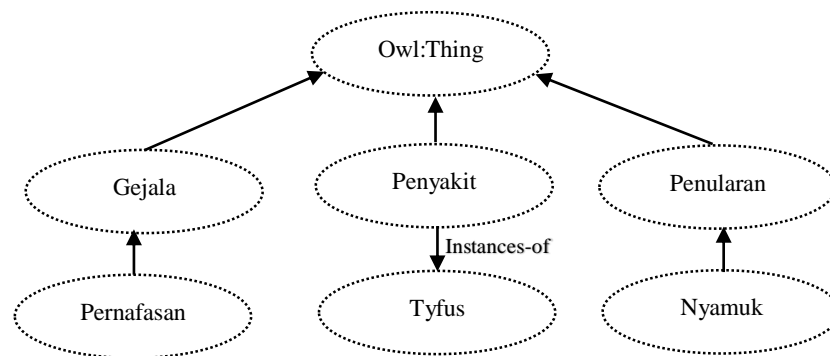
Sebelum proses semantik *ontology* dilakukan terlebih dahulu membuat sebuah model representasi beserta objek yang digunakan. Berikut representasinya ditunjukkan pada gambar 3.3.



Gambar 3.3 model representasi

Gambar diatas menunjukkan model representasi. Panah pada gejala menghubungkan ke penyakit dengan *object property* dimiliki, sedangkan panah pada penyakit terhubung ke gejala dengan *object property* memiliki. Panah pada penyakit terhubung ke penularan dengan *object property* disebabkan, dan penularan terhubung ke penyakit dengan *object property* menyebabkan.

Dengan representasi yang dibuatnya semantik ontologi akan mudah dibuat dan diselesaikan serta mudah dipahami. Representasi tersebut berupa *class* dimana yang paling umum dan *subclass* sebagai kekhususan pada *class*. Berikut contoh relasinya ditunjukkan pada gambar 3.4.



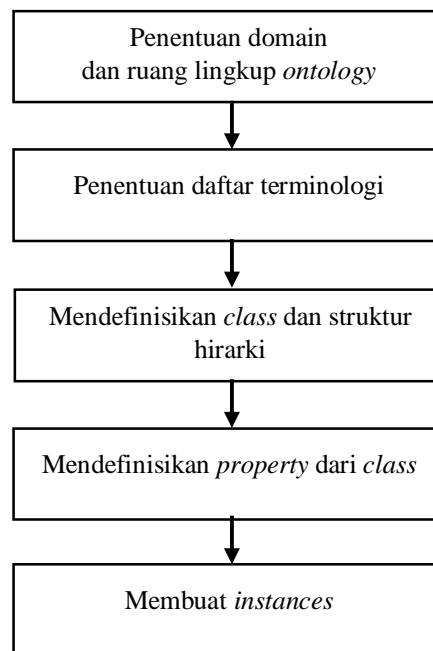
Gambar 3.4 Relasi class yang umum

Dari relasi tersebut ditunjukkan Gejala, Penyakit, Penularan adalah *class* yang memiliki kedudukan tertinggi sehingga berada diatas. Tyfus mempunyai relasi instances-of bahwa tyfus mejadi instances pada penyakit. Nyamuk dan pernafasan merupakan *subclass* atau kekhususan dari *class* diatasnya. Data dari gejala dan penularan dari penyakit yang diinginkan diperlukannya buku yang berkaitan, tenaga medis baik perawat maupun

dokter. Penambahan data properti pada sistem yang digunakan untuk menambah kelengkapan data informasi yang digunakan.

3.5 Perancangan Ontologi

Perancangan ontologi memberikan informasi mengenai tahapan-tahapan dalam pembangunan ontology dan menjelaskan mengenai komponen apa saja yang dibutuhkan dalam penggambaran sebuah informasi. Ada beberapa tahapan dalam penelitian perancangan semantik ontology. Tahapan tersebut dijelaskan pada gambar 3.5.



Gambar 3.5 Tahapan ontology

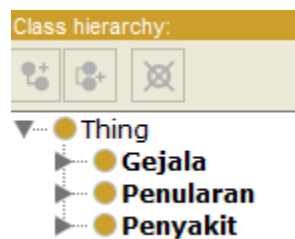
Menurut Noy dan McGuinness (2000) telah menjelaskan ada beberapa langkah-langkah yang harus diperlukan didalam pembangunan ontology, salah satunya dengan menentukan konsep dan domain.

3.5.1 Penentuan Konsep dan Domain

Penentuan konsep dan domain pada dasarnya mempresentasikan semua data dilengkapi informasi yang disusun berdasarkan klasifikasi dan dikelompokkan kedalam jenis-jenis yang sama.

Tahapan awal dalam membangun struktur *ontology* yaitu dengan menentukan ruang lingkup dan domain. Mempertimbangkan penggunaan *ontology* yang sudah ada. Menentukan daftar terminology, dan mendefinisikan *class* dan hirarkinya.

Domain ontologi yang akan dibangun yaitu informasi mengenai penyakit dari gejala yang dirasakan. *Ontology* ini digunakan untuk aplikasi yang memberikan informasi mengenai gejala yang dirasakan yang akan mendiagnosis kedalam suatu penyakit. Dapat dilihat pada gambar 3.6.



Gambar 3.6 Rancangan Pendefinisian Class

3.5.2 Definisi Kelas dan Struktur Hirarki

Definisi kelas adalah pengelompokan kelas-kelas dengan karakteristik yang sama yang muncul dalam sebuah domain. Uschold dan Gruninger (1996) menegaskan didalam makalah “*Ontologies: Principles, Methods and Applications*” bahwa ada beberapa pendekatan metode yang dapat digunakan untuk membangun struktur hirarki kelas diantaranya yaitu metode *top-down*, *bottom-up*, *combination*. Untuk aplikasi diagnosis penyakit ini digunakan pendekatan *top-down*, dimana kelas-kelas didefinisikan dari mulai konsep yang paling umum sampai konsep yang lebih spesifik. Rancangan kelas hirarki dapat dijelaskan pada gambar 3.7.



Gambar 3.7 Rancangan Struktur Hirarki Kelas Utama

- Gejala, pengelompokan gejala pada penyakit tertentu
- Penyakit, pengelompokan penyakit yang dibutuhkan
- Penularan, pengelompokan penularan pada penyakit tertentu

Tahapan selanjutnya adalah penentuan *subclass* yang berfungsi untuk mengkhususkan *class* umum. Dapat dilihat pada gambar 3.8.

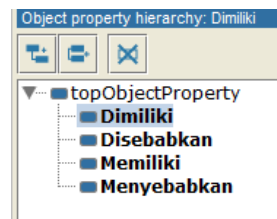


Gambar 3.8 Rancangan Pendefinisian Subclass

- a. *Class* Gejala, *subclass* Darah, Eksresi, Kelenjar, Kepala, Kulit, Mata, Otot, Pencernaan, Pernafasan, Perut, Punggung, Sendi, Telinga, Tubuh, Tulang, Leher, Dada, Wajah
- b. *Class* Penyakit, *subclass* Darah, Saluran Pernafasan, Saluran Pencernaan, Saraf
- c. *Class* Penularan, Air Tercemar, Kuman, Makanan, Nyamuk, Pakaian, Udara, Virus, Bakteri

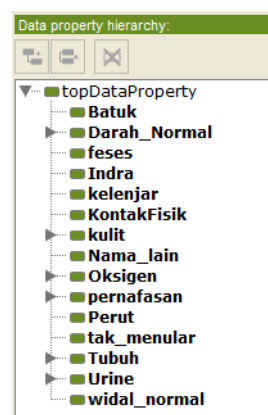
3.5.3 Mendefinisikan Properties (Slot)

Mendefinisikan property dari kelas yaitu dengan mengisikan nilai slot pada *object properties* dan *data properties*. Berikut rancangan *object properties* untuk diagnosis penyakit menggunakan semantik ontologi pada Gambar 3.9 dan *data properties* pada Gambar 3.10.



Gambar 3.9 Rancangan *Object Properties*

Gambar diatas menunjukkan perancangan *object property* yang dibuat untuk menghubungkan *instance* dengan *instance lainnya*.

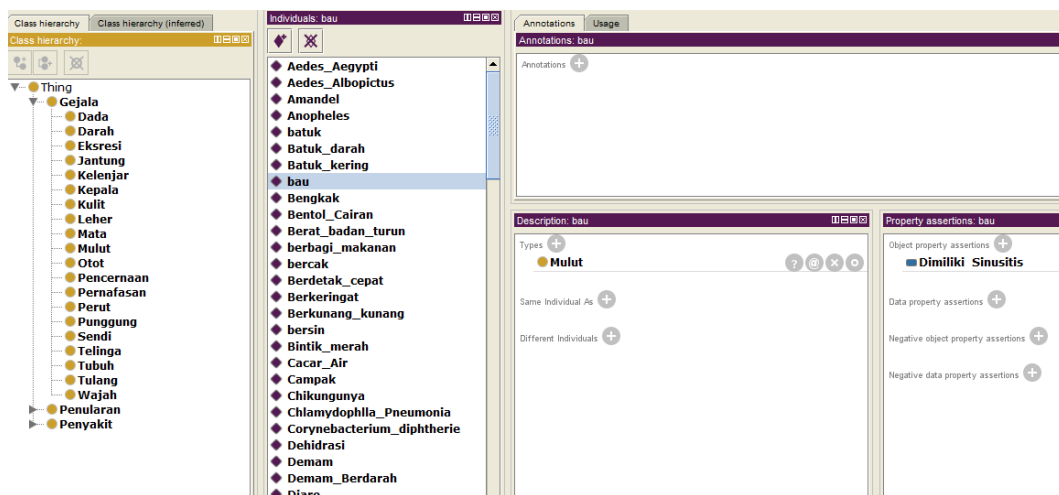


Gambar 3.10 Rancangan *Data Property*

Gambar diatas menunjukkan perancangan *data property* yang dibuat untuk menghubungkan *instance* dengan *type datanya*.

3.5.4 Mendefinisikan Instances – instances

Mendefinisikan *instances* yaitu pengisian nilai-nilai pada slot di setiap kelas. *Instances* merupakan komponen yang penting dalam *semantic ontology*, dengan pengisian *instances* aplikasi akan memiliki value yang ditampilkan pada saat diquery dengan SPARQL. Contoh perancangan instances pada *class* gejala *subclass* tubuh dan pencernaan ditunjukkan pada gambar 3.11.



Gambar 3.11 Rancangan Instances

Gambar diatas menunjukkan perancangan *instance*, dimana *instance* tersebut akan saling terhubung dengan *object property*. Dan saat saya memilih salah satu individual yaitu bau terdapat description mulut, menunjukkan *instance* tersebut milik class gejala dan subclass mulut. Pada *property assertions* dimiliki sinusitis, bahwa *instance* tersebut terhubung dengan *instance* sinusitis yang dihubungkan dengan *object property* dimiliki.

3.6 Query Jena Fuseki

Salah satu tahapan penting dalam pencarian diagnosis yang akan dibangun adalah tahap *query* di server Jena Fuseki. Jena menyediakan

perpustakaan Java yang luas untuk membantu *developer* mengembangkan kode yang menangani RDF, RDFS, RDFa, OWL dan SPARQL sesuai dengan rekomendasi dari W3C yang diterbitkan. Jena juga menyertakan *rule-based inference engine* untuk melakukan penalaran berdasarkan ontologi OWL. Berikut contoh query dari proses pencarian:

```

1
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX owl: <http://www.w3.org/2002/07/owl#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 PREFIX uni: <http://www.semanticweb.org/mpc/ontologies/2018/11/untitled-ontology-41>
6 PREFIX uni: <http://www.gejalapenyakit.com/ontologies/2018/8/disertaidemam#>
7
8 SELECT ?s ?p ?penyakit
9 WHERE {
10   uni:Demam ?p ?penyakit .
11   uni:Ruam ?p ?penyakit .
12
13
14 }

```

Gambar 3.12 Contoh Query Jena Fuseki

Query di atas adalah salah satu contoh query untuk melakukan pencarian berdasarkan subjek, predikat, dan objek dengan limit yang ditentukan. Pada *query* tersebut terdapat *namespace prefix* yang terikat (*binding*). Berikut contoh hasil dari *query* di atas:

s	p	penyakit
1	rdf:type	owl:NamedIndividual
2	uni:Dimiliki	uni:Meningitis
3	uni:Dimiliki	uni:Chikungunya
4	uni:Dimiliki	uni:Zika
5	uni:Dimiliki	uni:Rubella
6	uni:Dimiliki	uni:Demam_Berdarah
7	uni:Dimiliki	uni:ISPA
8	uni:Dimiliki	uni:Tytus

Gambar 3.13 Hasil Query Jena Fuseki

Didapatkan hasil dari *query* adalah data yang ada dan saling berkesinambungan di dalam ontologi yang dimodelkan dengan limit yang telah ditentukan ditampilkan.

3.7 Perancangan Interface

Rancangan desain dari *interface* ini dibuat berdasarkan kebutuhan sistem untuk memberikan sajian yang menarik, mudah dioperasikan oleh *user*.

1. Halaman Utama

Halaman menu utama merupakan halaman awal ketika semantik ontologi telah dibuat dan diquery dengan SPARQL. Pada halaman ini menampilkan beberapa menu gejala yang harus dipilih oleh pengguna yang merasakan gejala tersebut untuk mengetahui diagnosi sebuah penyakit. Dapat dilihat pada gambar 3.14.

Gejala yang dirasakan			
Gejala 1	<input type="checkbox"/>	Gejala 2	<input checked="" type="checkbox"/>
Gejala 3	<input checked="" type="checkbox"/>	Gejala 4	<input type="checkbox"/>
Gejala 5	<input type="checkbox"/>	Gejala 6	<input type="checkbox"/>
Gejala 7	<input checked="" type="checkbox"/>	Gejala 8	<input type="checkbox"/>
Gejala 9	<input checked="" type="checkbox"/>	Gejala 10	<input checked="" type="checkbox"/>
Gejala 11	<input checked="" type="checkbox"/>	Gejala 12	<input checked="" type="checkbox"/>

Gambar 3.14 Rancangan Tampilan Menu Utama

2. Halaman Tampilan Penyakit dan Penularan

Pada halaman ini akan menampilkan penyakit yang muncul dari gejala yang sudah dipilih beserta penularan yang mengakibatkan penyakit tersebut. Dapat dilihat rancangannya pada gambar 3.15.

Sesuai dengan tabel setelah nama dan umur responden terisi, responden perlu memilih beberapa gejala. Sistem akan memberikan hasil diagnosis dari gejala tersebut. Relitasnya apakah sama dengan ontologi yang telah dibuat atau tidak. Agar diagnosis yang diperlukan tepat, dibutuhkannya ahli dalam bidang tersebut. Hasilnya berupa TP, TF, FP, FN, agar dapat dihitung akurasi.

3.9 Kebutuhan Pembangunan Sistem

Dalam pembuatan semantik *ontology* yang telah dirancang membutuhkan perangkat lunak (*Software*) dan perangkat keras (*Hardware*). Kebutuhan perangkat keras serta perangkat lunak dari sistem sebagai berikut:

3.9.1 Perangkat Lunak (Software)

1. Sistem Operasi Windows 7/8/8.1/10
2. Protege
3. Server Apache Jena Fuseki 3.9.0.
4. XAMPP v3.1.0

3.9.2 Perangkat Keras (Hardware)

1. Processor Intel Pentium Core i3 1.9 Ghz
2. Memory RAM 2 GB
3. Monitor VGA atau SVGA 14 inch
4. Harddisk 500 GB
5. Keyboard dan Mouse