

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Setiap makhluk hidup membutuhkan makanan. Mengonsumsi makanan yang baik dan sehat akan membantu pertumbuhan dan perkembangan tubuh. Namun, saat ini banyak sekali jenis makanan yang beredar di masyarakat. Tidak semua makanan tersebut baik bagi kesehatan jika dikonsumsi. Terdapat resiko terkena penyakit yang tidak diinginkan jika mengonsumsi makanan secara sembarangan. Penyakit yang diderita selama ini akan kambuh kembali jika tidak memerhatikan bahan-bahan yang dipakai dalam mengolah makanan tersebut. Nilai gizi dan bahan dapat dijadikan tolak ukur dalam menentukan makanan yang baik bagi kesehatan. Maka dari itu, resep yang memiliki nilai gizi dan bahan yang sesuai tidak akan berakibat buruk dan akan sangat bermanfaat bagi kesehatan.

3.2. Hasil Analisis

Hasil analisis masalah yang didapatkan bahwa perlunya sistem yang dapat menentukan resep makanan yang sesuai dengan keadaan *user*. Dengan memasukkan jenis kelamin, usia, dan riwayat penyakit yang diderita ke dalam portal, sistem akan memproses melalui model *ontology* yang telah dibuat oleh *domain expert*, kemudian hasilnya akan ditampilkan ke dalam portal berupa pilihan resep yang direkomendasikan oleh sistem.

Sebagai bagian dari persyaratan sistem dan kegiatan perancangan, terdapat beberapa komponen yang berkaitan, yaitu:

1. **Komponen Rancangan Model *Ontology***

Komponen ini digunakan untuk merancang model *ontology* berdasarkan *domain expert*. Dalam pemodelan *ontology* memanfaatkan sebuah aplikasi *open-source* Protege. Hasil rancangan berupa model *ontology* yang tersusun hirarki taksonomi (*classes - subclasses*). *Domain expert* membuat hirarki taksonomi, *property*, dan *instance* dengan menggunakan Protege. Setelah

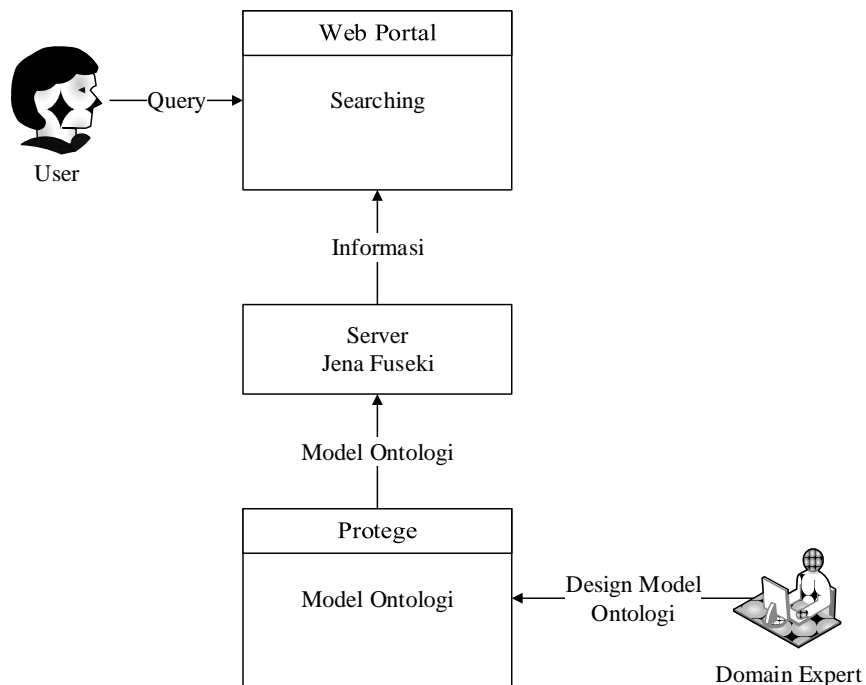
model *ontology* selesai dibuat, maka model *ontology* tersebut disimpan dalam *file* berekstensi OWL.

2. Komponen Server

Komponen ini digunakan oleh *domain expert* untuk *upload* dan *parsing file* OWL pada server Jena Fuseki untuk kebutuhan query.

3. Komponen Portal

Komponen ini dibuat dengan tujuan sebagai perantara untuk *user* sebagai wadah informasi yang diinginkan.



Gambar 3.1. Skema Keseluruhan Sistem

Pembangunan sistem dimulai dengan perancangan model *ontology* dengan mengimplementasikan *class*, *properties*, dan *instance* melalui Protege oleh *domain expert*. Dilanjutkan dengan *upload* dan *parsing file* OWL pada server Jena Fuseki. Selanjutnya pembuatan portal *semantic web* sebagai perantara informasi kepada *user*. Portal tersebut memuat query yang akan dikirimkan pada server Jena Fuseki. Jena Fuseki memproses query yang dikirimkan dan mengembalikan informasi pada portal hasil *searching* yang dilakukan oleh *user*.

3.3. Representasi Model

Model *ontology* dibuat menggunakan *tool* protege dan dijalankan melalui jena fuseki server. Proses pembuatan model melibatkan resep makanan yang dihubungkan dengan tabel angka kecukupan gizi dan bahan-bahan yang merupakan bahan anjuran dan pantangan bagi penderita penyakit tertentu.

3.3.1. Perancangan *Ontology*

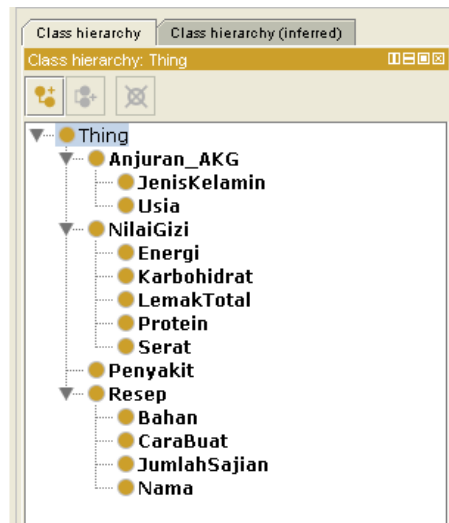
Perancangan *ontology* memberikan informasi mengenai tahapan-tahapan dalam pembangunan *ontology* dan menjelaskan mengenai komponen apa saja yang dibutuhkan dalam penggambaran sebuah informasi.

Tahapan-tahapan dalam pembangunan *ontology* adalah sebagai berikut:

1. Penentuan domain. Domain yang melingkupi *ontology* ini adalah resep makanan, angka kecukupan gizi, dan bahan anjuran/pantangan bagi penderita penyakit tertentu.
2. Mendefinisikan *ontology class* dan menyusun *class* tersebut dalam hirarki taksonomi (*subclass-superclass*) dengan menggunakan proses pengembangan *top-down* dimulai dengan mendefinisikan konsep umum dalam domain dilanjutkan dengan konsep yang lebih spesifik.
3. Mendefinisikan *slot* atau *property*
4. Mendefinisikan *facets* pada *slot* atau *axiom* pada *properties*. *Properties* memiliki *domain* dan *range* yang spesifik. *Properties* menghubungkan individu pada *domain* dan individu pada *range*.
5. Membuat *instance*.
6. Mengisi nilai *slot* pada *instance*.

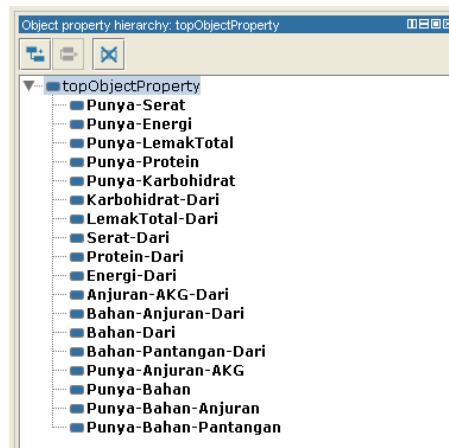
3.3.2. Pembangunan dalam Protege

Model *ontology* yang telah dirancang dilanjutkan dengan memvisualisasikan menggunakan protege, sebuah *open source platform*.



Gambar 3.2. Class pada Protege

Perancangan *ontology* untuk menyusun *class* dalam bentuk hirarki taksonomi (*subclass-superclass*) dapat dilakukan pada kolom “Class Property Hierarchy” pada Protege.



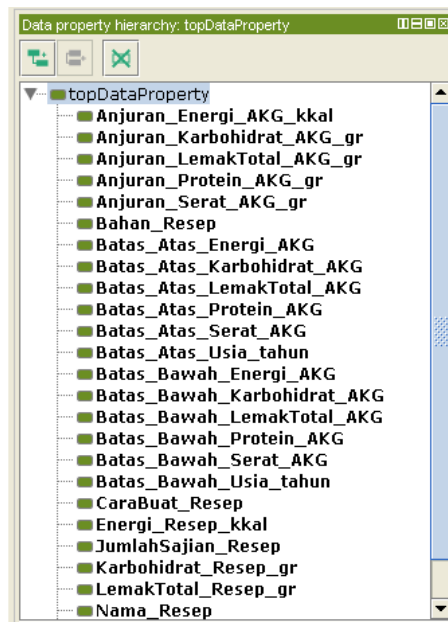
Gambar 3.3. Object Properties pada Protege

Tabel 3.1. Karakteristik dalam *Object Properties*

Karakteristik	Subjek	Predikat	Objek
Functional	Resep	Punya-Bahan	Bahan
Inverse Functional	Bahan	Bahan-Dari	Resep
Functional	Resep	Punya-Energi	Energi

Inverse Functional	Energi	Energi-Dari	Resep
Functional	Resep	Punya-Karbohidrat	Karbohidrat
Inverse Functional	Karbohidrat	Karbohidrat-Dari	Resep
Functional	Resep	Punya-LemakTotal	LemakTotal
Inverse Functional	LemakTotal	LemakTotal-Dari	Resep
Functional	Resep	Punya-Protein	Protein
Inverse Functional	Protein	Protein-Dari	Resep
Functional	Resep	Punya-Serat	Serat
Inverse Functional	Serat	Serat-Dari	Resep
Functional	NilaiGizi	Punya-Anjuran-AKG	Anjuran_AKG
Inverse Functional	Anjuran_AKG	Anjuran-AKG-Dari	NilaiGizi
Functional	Penyakit	Punya-Bahan-Anjuran	Bahan
Inverse Functional	Bahan	Bahan-Anjuran-Dari	Penyakit
Functional	Penyakit	Punya-Bahan-Pantangan	Bahan
Inverse Functional	Bahan	Bahan-Pantangan-Dari	Penyakit

Perancangan *ontology* untuk menyusun *facets* pada *slot* atau *axiom* dapat dilakukan pada kolom “Object Property Hierarchy” pada Protege.

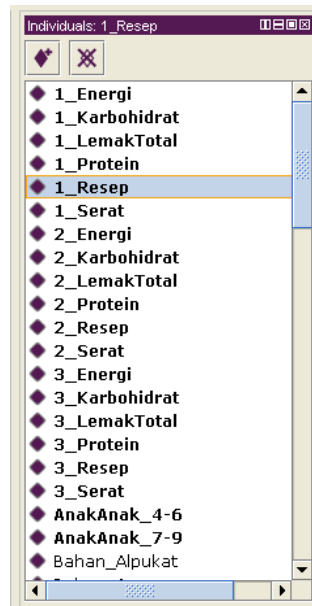


Gambar 3.4. *Data Properties* pada Protege

Tabel 3.2. Karakteristik dalam *Data Properties*

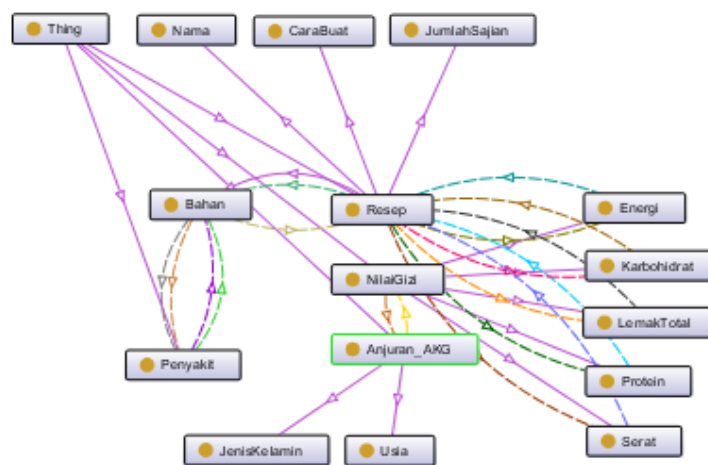
Karakteristik	Subjek	Predikat	Objek
Functional	Resep	Nama_Resep	Nama
Functional	Bahan	Bahan-Resep	Resep
Functional	Resep	CaraBuat_Resep	CaraBuat
Functional	Resep	JumlahSajian_Resep	JumlahSajian
Functional	Energi	Energi_Resep_kkal	Resep
Functional	Karbohidrat	Karbohidrat_Resep_gr	Resep
Functional	LemakTotal	LemakTotal_Resep_gr	Resep
Functional	Protein	Protein_Resep_gr	Resep
Functional	Serat	Serat_Resep_gr	Resep
Functional	Anjuran_AKG	Anjuran_Energi_AKG_kkal	Energi
Functional	Anjuran_AKG	Batas_Bawah_Energi_AKG	Energi
Functional	Anjuran_AKG	Batas_Atas_Energi_AKG	Energi
Functional	Anjuran_AKG	Anjuran_Karbohidrat_AKG_kkal	Karbohidrat
Functional	Anjuran_AKG	Batas_Bawah_Karbohidrat_AKG	Karbohidrat
Functional	Anjuran_AKG	Batas_Atas_Karbohidrat_AKG	Karbohidrat
Functional	Anjuran_AKG	Anjuran_LemakTotal_AKG_kkal	LemakTotal
Functional	Anjuran_AKG	Batas_Bawah_LemakTotal_AKG	LemakTotal
Functional	Anjuran_AKG	Batas_Atas_LemakTotal_AKG	LemakTotal
Functional	Anjuran_AKG	Anjuran_Protein_AKG_kkal	Protein
Functional	Anjuran_AKG	Batas_Bawah_Protein_AKG	Protein
Functional	Anjuran_AKG	Batas_Atas_Protein_AKG	Protein
Functional	Anjuran_AKG	Anjuran_Serat_AKG_kkal	Serat
Functional	Anjuran_AKG	Batas_Bawah_Serat_AKG	Serat
Functional	Anjuran_AKG	Batas_Atas_Serat_AKG	Serat
Functional	Usia	Batas_Atas_Usia_tahun	Anjuran_AKG
Functional	Usia	Batas_Bawah_Usia_tahun	Anjuran_AKG

Perancangan *ontology* untuk menyusun pada *slot* atau *property* dapat dilakukan pada kolom “Data Property Hierarchy” pada Protege.



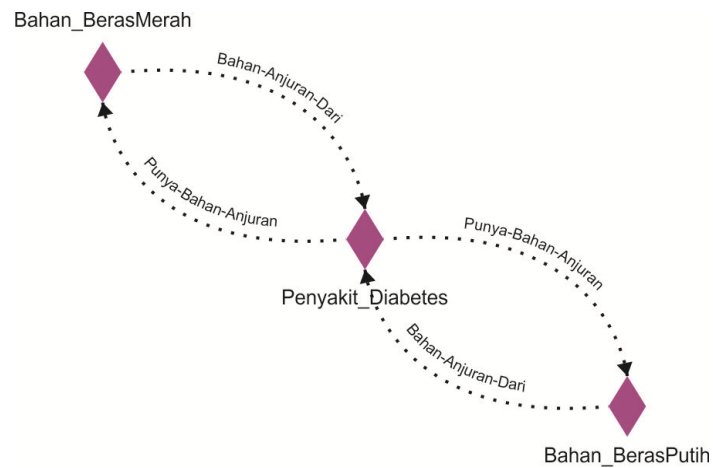
Gambar 3.5. *Individuals* pada Protege

Perancangan *ontology* untuk menyusun pada *instance* dan mengisi nilai dapat dilakukan pada kolom “Individuals” pada Protege. Pada kolom “Individuals” juga dapat digunakan untuk menempatkan data yang dibutuhkan. Data resep, tabel AKG, dan bahan-bahan yang terkait dengan penyakit dimasukkan ke dalam setiap kolom. Kemudian data tersebut dapat dipanggil dan digunakan menggunakan query SPARQL yang disambungkan dengan *jena fuseki server*.



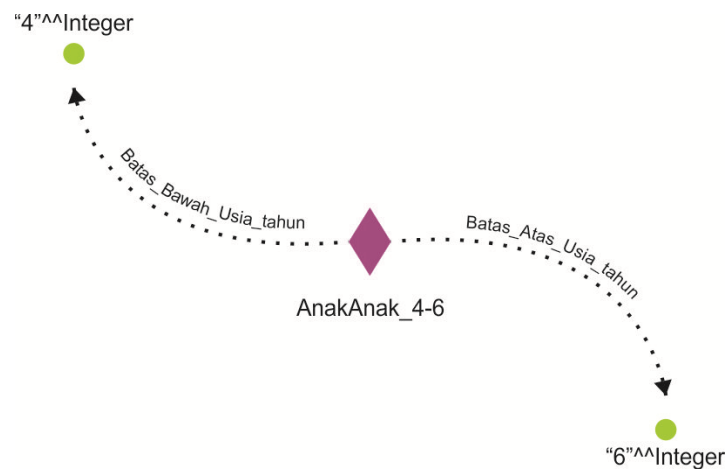
Gambar 3.6. Model *Ontology* Resep Makanan Sehat

Hasil bentuk model perancangan *ontology* dapat dilihat pada kolom “Ontograf” pada Protege dengan memasukkan *class* ke dalam *drawer* yang telah dibangun.



Gambar 3.7. Model Individual Penyakit

Salah satu *individual* dari *class* “Penyakit” yaitu “Penyakit_Diabetes”. Model perancangan tersebut dapat dilihat pada Gambar 3.7.



Gambar 3.8. Model Individual Anjuran AKG

Salah satu *individual* dari *class* “Anjuran_AKG” yaitu “AnakAnak_4-6”. Model perancangan tersebut dapat dilihat pada Gambar 3.8.

3.3.3. Query pada Jena Fuseki

Salah satu tahapan penting dalam penentuan resep makanan sehat yang akan dibangun adalah tahapan query pada 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 *ontology* OWL. Berikut contoh query dari proses pencarian:

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 PREFIX uni: <http://www.semanticweb.org/khoirulwafiq/ontologies/2019/2/untitled-ontology-4#>
6
7 SELECT ?subject ?predicate ?object
8 WHERE {
9   ?subject ?predicate ?object
10 }

```

Gambar 3.9. Contoh Query Jena Fuseki

Query diatas adalah salah satu contoh query untuk melakukan pencarian berdasarkan *subject*, *predicate*, dan *object*. Pada query tersebut terdapat *namespace prefix* yang terikat (*binding*). Berikut contoh hasil dari query diatas:

20	uni:1_Resep	uni:Bahan_Resep	"2 buah pir, buang bijinya dan potong dadu 2 buah apel, buang bijinya dan potong dadu 120 gr buah strawberry, buang tangkainya dan belah dua 120 gr buah blueberry 2 buah jeruk, kupas, buang bijinya dan potong dadu 2 buah kiwi, kupas dan potong dadu 50 gr buah anggur putih tanpa biji 2 buah pisang matang, iris-iris 3 sdm sari buah tanpa pemanis atau perasan air jeruk segar"
21	uni:1_Resep	uni:Punya-Karbohidrat	uni:1_Karbohidrat
22	uni:1_Resep	uni:CaraBuat_Resep	"Siapkan dan potong semua buah sesaat sebelum disajikan. Campur potongan buah ke mangkuk, tuang sari buah atau jeruk, aduk hingga rata."
23	uni:1_Resep	rdfs:type	owl:NamedIndividual
24	uni:1_Resep	uni:Punya-LemakTotal	uni:1_LemakTotal
25	uni:1_Resep	uni:Punya-Energi	uni:1_Energi
26	uni:1_Resep	uni:Punya-Protein	uni:1_Protein
27	uni:1_Resep	uni>Nama_Resep	"SALAD BUAH SEGAR"

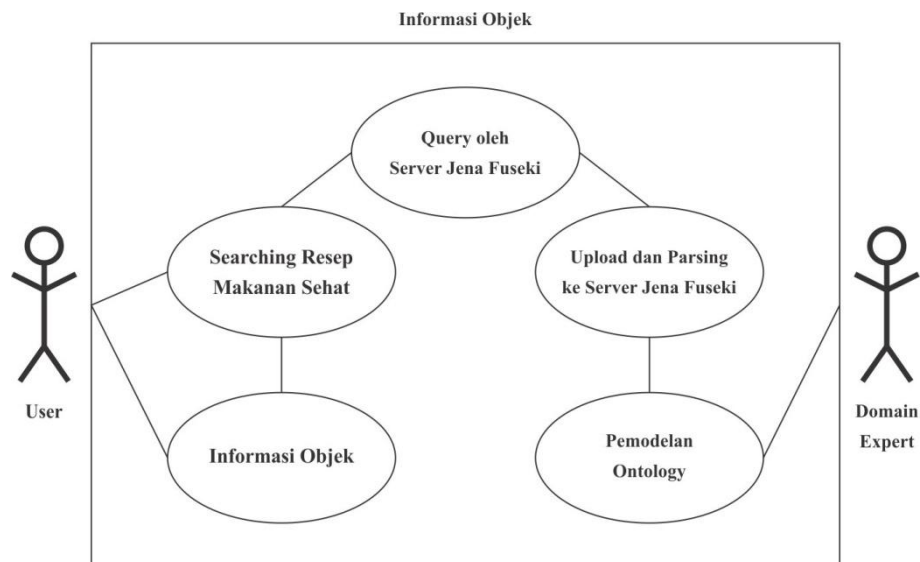
Gambar 3.10. Hasil Query Jena Fuseki

Didapatkan hasil dari query diatas adalah data yang ada dan saling berkesinambungan di dalam *ontology* yang dimodelkan.

3.4. Perancangan Sistem

Perancangan sistem merupakan tahapan setelah melakukan analisis dari pengembangan sistem, pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi dan gambaran bagaimana suatu sistem dapat terbentuk.

3.4.1. Use Case Diagram



Gambar 3.11. Use Case Diagram Resep Makanan Sehat

Berikut adalah penjelasan *use case* secara naratif bagaimana sistem akan berjalan:

1. Pencarian Resep Makanan Sehat

- a. Objective : Pencarian resep makanan sehat.
- b. Actors : *User*
- c. Pre-condition :

User menginginkan suatu informasi terhadap Resep makanan yang meliputi bahan-bahan, cara memasak, dan nilai gizi yang terkandung didalamnya

- d. Normal Flow :
 - 1) *User* membuka halaman web portal.
 - 2) *User* memilih informasi apa yang akan dicari.
 - 3) *User* memulai pencarian.
 - 4) *User* mendapatkan informasi mengenai resep makanan sehat dari pemilihan pencarian informasi yang dipilih oleh *user*.
- e. Alternate Flow :

User tidak mendapatkan informasi yang diinginkan atau informasi tersebut tidak terdapat dalam model *ontology* yang sudah dibangun.
- f. Post Condition : *User* memperoleh hasil pencarian.

2. Pemodelan *Ontology*

- a. Objective :

Membangun desain *ontology* dalam domain resep makanan sehat ke dalam *class*, *properties* dan *instance* dengan menggunakan *tool* Protege.
- b. Actors : Domain Expert
- c. Normal Flow :
 - 1) *Domain expert* melakukan pemodelan terhadap rancangan *ontology* menggunakan Protege dan disimpan dalam bentuk *file* OWL.
 - 2) *File* OWL di-*upload* dan di-*parsing* ke server Jena Fuseki.
- d. Result :

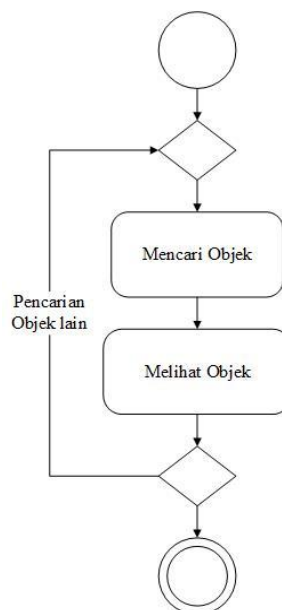
Model *ontology* terintegrasikan dengan server Jena Fuseki.

3.4.1.1. Activity Diagram



Gambar 3.12. Activity Diagram Domain Expert Resep Makanan Sehat

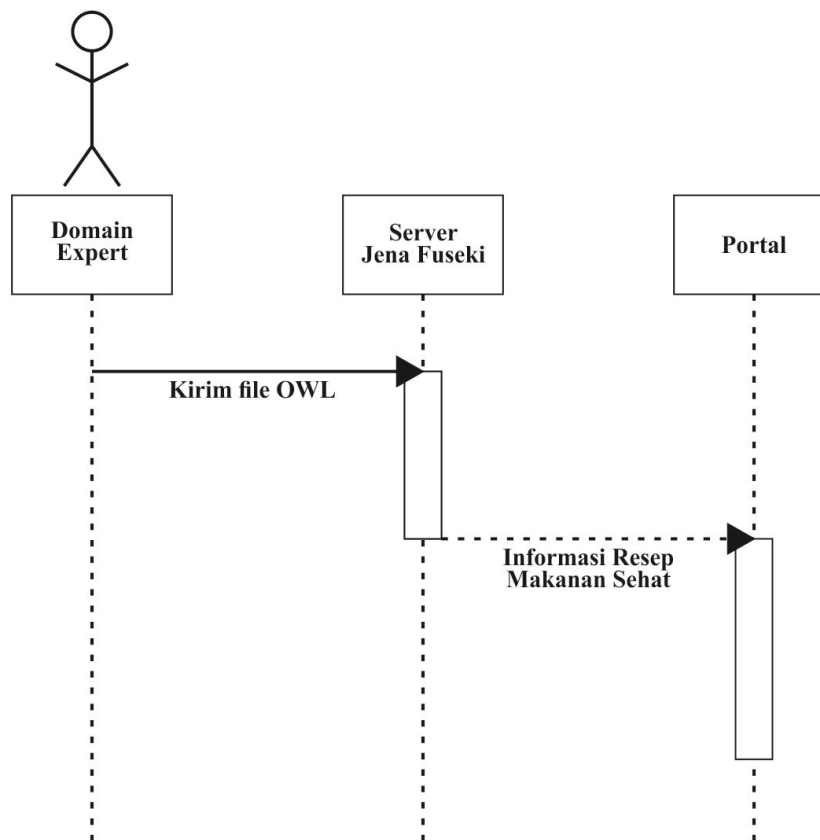
Activity Diagram pada Gambar 3.12 menjelaskan awal proses dari seorang *Domain Expert* dengan memodelkan *ontology*. Model *ontology* tersebut divisualisasikan ke dalam sebuah *file OWL* melalui aplikasi *open source platform*, Protege dan kemudian di *upload* dan *parsing* ke server Jena Fuseki.



Gambar 3.13. Activity Diagram User Resep Makanan Sehat

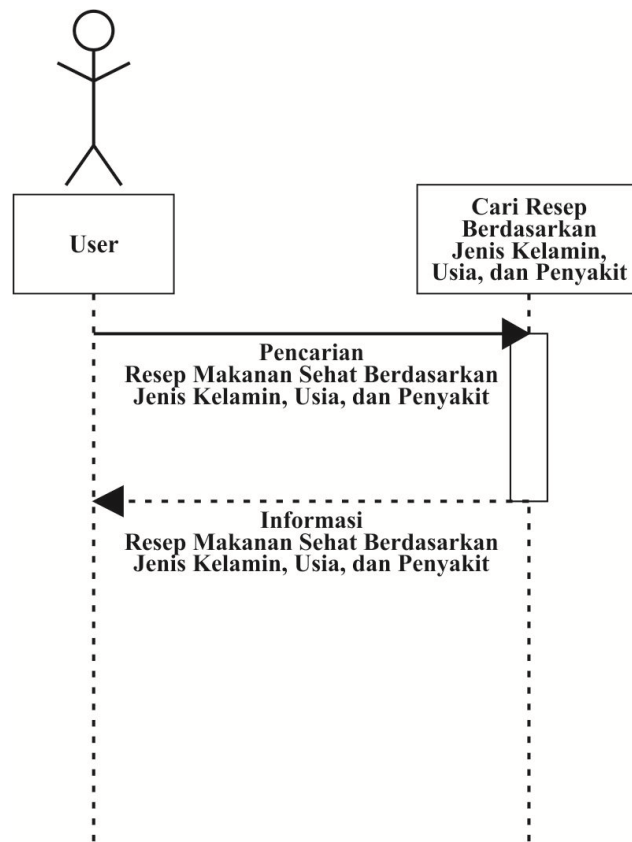
Activity Diagram pada Gambar 3.13 menjelaskan awal proses dari *user* dengan melakukan pencarian objek. Objek yang dicari akan ditampilkan kepada *user*, tetapi *user* dapat melakukan pencarian terhadap objek lainnya yang saling berkaitan.

3.4.1.2. *Interaction Diagram*



Gambar 3.14. *Interaction Diagram Domain Expert Resep Makanan Sehat*

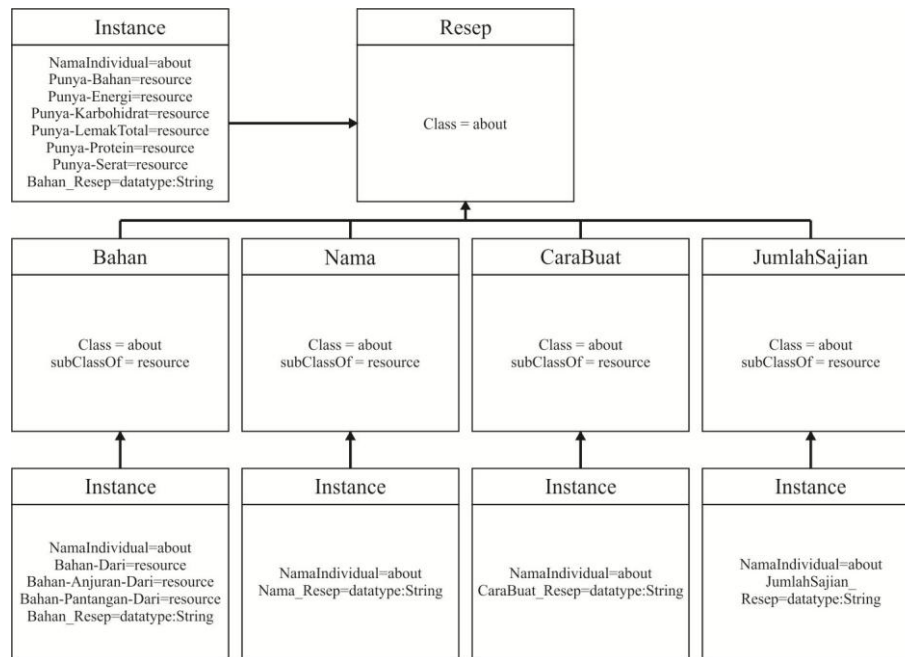
Interaksi Diagram pada Gambar 3.14 menjelaskan interaksi yang dilakukan oleh seorang *Domain Expert* dengan mengirimkan *file* OWL yang telah divisualisasikan pada Protege ke server Jena Fuseki. Server Jena Fuseki mengembalikan informasi kepada portal.



Gambar 3.15. *Interaction Diagram User Resep Makanan Sehat*

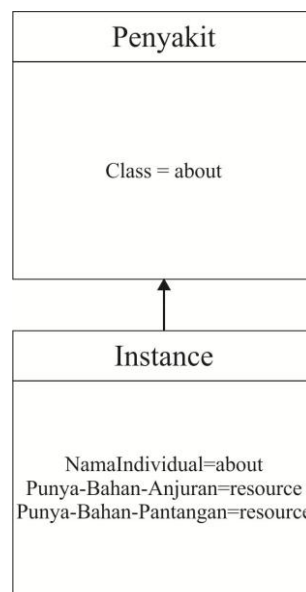
Interaksi Diagram pada Gambar 3.15 menjelaskan interaksi yang dilakukan oleh *User* dengan melakukan pencarian resep makanan sehat berdasarkan jenis kelamin, usia dan penyakit. *User* mendapatkan kembali informasi yang dicari berupa bahan-bahan, cara memasak, dan nilai gizi yang terkandung dalam resep makanan sehat.

3.4.1.3. Class Diagram



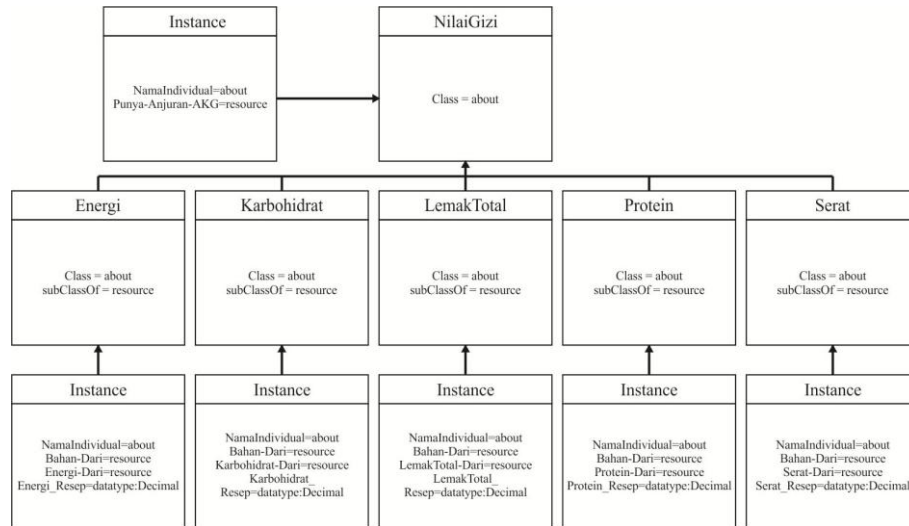
Gambar 3.16. Class Resep

Class Resep yang memiliki *instance* dan *subclass* yang terdiri dari Bahan, Nama, CaraBuat, dan JumlahSajian. Setiap *subclass* tersebut terdiri dari *instance* seperti pada gambar 3.16.



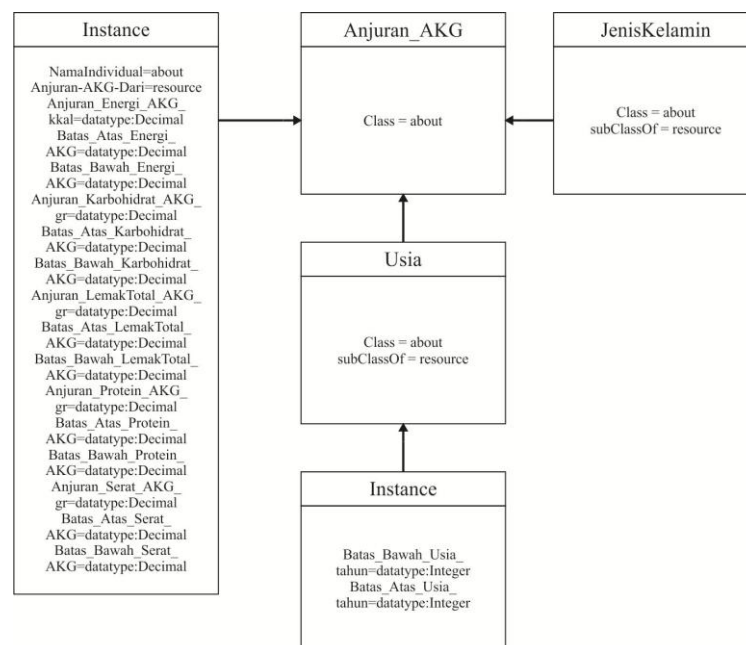
Gambar 3.17. Class Penyakit

Class Penyakit terdiri dari *instance* seperti pada gambar 3.17.



Gambar 3.18. Class NilaiGizi

Class NilaiGizi yang memiliki *instance* dan *subclass* yang terdiri dari Energi, Karbohidrat, LemakTotal, Protein, dan Serat. Setiap *subclass* tersebut terdiri dari *instance* seperti pada gambar 3.18.



Gambar 3.19. Class Anjuran_AKG

Class Anjuran_AKG terdiri dari *instance* seperti pada gambar 3.19.

Berdasarkan Tabel 2.1, angka yang ada di tabel AKG merupakan nilai gizi setiap orang per hari. Jika diasumsikan setiap orang makan tiga kali sehari, angka tersebut harus dibagi tiga agar dapat digunakan.

Kata “kecukupan” berasal dari kata dasar “cukup”. Menurut Kamus Besar Bahasa Indonesia (ApaArti, 2019), kata “cukup” berarti tidak kurang. “Cukup” juga berarti sudah memadai (tidak perlu ditambah lagi). Jika angka yang digunakan tepat seperti apa yang ada di tabel AKG, akan sulit mencari resep yang sesuai dengan angka tersebut. Maka dari itu, angka yang digunakan berada diantara angka batas atas dan batas bawah. Angka batas atas dan batas bawah masing-masing ditentukan dengan menjumlah dan mengurangi angka yang ada di tabel AKG dengan nilai titik tengah antara angka yang ada di tabel AKG dengan angka dari kelompok umur di sekitarnya. Berdasarkan ketentuan tersebut, batas atas dan batas bawah nilai gizi yang ada di tabel AKG dapat diuraikan seperti Tabel 3.3.

Tabel 3.3. Batas Atas dan Batas Bawah Nilai Gizi AKG

Kelompok Umur		Batas	Energi	Karbohidrat	Lemak Total	Protein	Serat
			(kkal)	(gr)	(gr)	(gr)	(gr)
Anak-Anak	4-6 tahun	Batas Bawah	454.166	62.5	17.666	10.166	6.333
		Batas Atas	575	79	22.333	14	8
	7-9 tahun	Batas Bawah	575	79	22.333	14	8
		Batas Atas	658.333	90.5	25.333	17.5	9.333
Laki-Laki	10-12 tahun	Batas Bawah	658.333	90.5	21.166	17.5	9.333
		Batas Atas	762.5	104.833	25.5	21.333	10.833
	13-15 tahun	Batas Bawah	762.5	104.833	25.5	21.333	10.833
		Batas Atas	858.333	118	28.666	23	12
	16-18 tahun	Batas Bawah	858.333	118	28.666	21.333	12
		Batas Atas	900	123.833	30	23	12.5
	19-29 tahun	Batas Bawah	900	123.833	30	20	12.5
		Batas Atas	916.666	128.166	30.666	21.167	12.833
	30-49 tahun	Batas Bawah	825	128.166	23	21.166	11.833
		Batas Atas	891.666	134.5	27.333	22.166	12.833

	50-64 tahun	Batas Bawah	704.166	109.666	19.666	21.166	10	
		Batas Atas	825	123.833	23	22.166	11.833	
	65-80 tahun	Batas Bawah	570.833	92.833	15.833	20.833	8.166	
		Batas Atas	704.166	109.666	19.666	21.166	10	
	80+ tahun	Batas Bawah	445.833	72.5	12.167	19.166	6.5	
		Batas Atas	570.833	92.833	15.833	20.833	8.166	
	Perempuan	10-12 tahun	Batas Bawah	641.666	88.166	21.666	18.166	9
			Batas Atas	687.5	94.5	23	21.5	9.666
		13-15 tahun	Batas Bawah	687.5	94.5	23	21.5	9.666
			Batas Atas	729.166	100.166	24.333	24.5	10.333
16-18 tahun		Batas Bawah	687.5	94.5	23	19.166	9.666	
		Batas Atas	729.166	100.166	24.333	21.333	10.333	
19-29 tahun		Batas Bawah	729.166	100.166	24.333	18.166	10.333	
		Batas Atas	770.833	105.833	25.666	19.166	11	
30-49 tahun		Batas Bawah	675	105.833	18.833	18.583	9.666	
		Batas Atas	743.75	109.5	22.833	19.416	10.5	
50-64 tahun		Batas Bawah	575	89.5	16	18.583	8.333	
		Batas Atas	675	102.25	18.833	19.416	9.666	
65-80 tahun		Batas Bawah	495.833	80.666	9	18.5	7	
		Batas Atas	575	89.5	16	18.833	8.333	
80+ tahun		Batas Bawah	454.166	74	0.333	18.166	6.333	
		Batas Atas	495.833	80.666	9	18.5	7	

3.5. Perancangan Portal

Portal web adalah situs web yang dibangun dengan menyediakan kemampuan tertentu dalam menyajikan informasi yang diinginkan kepada para penggunanya. Berikut adalah rancangan portal yang akan dibangun:

Gambar 3.20. Halaman Form

Halaman utama berisi tentang form yang akan diisi oleh *user*. *User* memasukkan data berupa jenis kelamin, usia, dan penyakit. Form jenis kelamin berupa jenis form *check list* yang berisi pilihan “Laki-Laki” atau “Perempuan”. Untuk form usia diisi dengan angka dengan ketentuan usia minimal 4 tahun. Sedangkan untuk form penyakit, berupa form *check list* yang berisi pilihan “Tidak Ada”, “Diabetes”, “Kolestrol”, “Hipertensi”, “Asam Urat”, atau “Penyakit Kulit”. Untuk form penyakit boleh tidak diisi, jika *user* tidak memiliki riwayat penyakit apapun. Dan *user* juga bisa tidak mengisi form jenis kelamin dan usia jika ingin mencari resep makanan yang sesuai dengan riwayat penyakit saja.

Gambar 3.21. Halaman Resep

Setelah mengisi form pada halaman utama seperti Gambar 3.20, daftar resep akan muncul sesuai dengan apa yang telah dimasukkan. Terdapat tombol “Reset” jika *user* ingin kembali ke halaman utama dan mengisi form yang berbeda seperti pada Gambar 3.20. Jika *user* mengklik nama resep yang ada pada halaman daftar resep, akan muncul tampilan seperti pada Gambar 3.22.

The screenshot shows a web interface titled "Resep Makanan Sehat" with a "Reset" button in the top right corner. The form contains the following fields:

Nama_Resep.....	NilaiGizi:
Bahan_Resep.....	Energi
..... kkal
.....	Karbohidrat
..... gr
.....	LemakTotal
..... gr
CaraBuat_Resep.....	Protein
..... gr
.....	Serat
..... gr
JumlahSajian_Resep	

Gambar 3.22. Halaman Keterangan Resep

Halaman Resep makanan berisi tentang resep secara keseluruhan, yaitu nama resep, bahan-bahan resep, cara memasak, jumlah sajian, dan nilai gizi yang terkandung dalam resep makanan. Di halaman resep makanan juga terdapat tombol “Reset” jika *user* ingin kembali ke halaman utama dan mengisi form yang berbeda seperti pada Gambar 3.20.

3.6. Evaluasi Sistem

Pada penelitian ini, untuk mengukur evaluasi kerja query *ontology* yang dibangun digunakan pengujian *precision* dan *recall* menggunakan *Confusion Matrix*. *Confusion matrix* menurut Han dan Kamber (2011) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-*

Negative memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data (Fibrianda, 2018).

Tabel 3.4. *Confusion Matrix*

Keterangan	Relevan	Tidak Relevan
Terambil	True Positive (TP)	False Positive (FP)
Tidak terambil	False Negative (FN)	True Negative (TN)

Rumus untuk menghitung *Precision* :

$$Precision = \frac{tp}{tp+fp} * 100\% \dots \dots \dots (3.2)$$

Rumus untuk menghitung *Recall* :

$$Recall = \frac{tp}{tp+fn} * 100\% \dots \dots \dots (3.3)$$

Nilai *precision* dan *recall* dinyatakan dalam persen. Semakin tinggi nilai tersebut menunjukkan semakin baiknya kinerja sistem. Evaluasi yang akan dilakukan dalam penelitian ini adalah menghitung nilai dari *precision* dan *recall* berdasarkan data yang berhasil dihasilkan oleh query yang dibuat. Sedangkan untuk menentukan nilai dari *precision* dan *recall* harus didapatkan jumlah data yang relevan terhadap suatu objek.

Mengukur *precision* dan *recall* perlu dilakukan penelitian kualitatif dari ahli dan koresponden agar sistem dapat dievaluasi dengan benar. Penelitian kualitatif menggunakan data kualitatif seperti wawancara, data observasi partisipan dan data dokumen untuk memahami dan menjelaskan fenomena sosial (Azmi, 2018). Maka dari itu, penulis bekerjasama dengan seorang ahli dan 10 koresponden untuk mengisi kuisisioner yang akan digunakan untuk proses evaluasi sistem.

3.7. Spesifikasi Pembangunan Sistem

Kebutuhan perangkat keras serta perangkat lunak dari sistem sebagai berikut:

1. Kebutuhan Perangkat Keras
 - a. CPU Intel N4000 setara atau lebih tinggi
 - b. RAM 2Gb setara atau lebih
 - c. HDD 500Gb setara atau lebih
 - d. Mouse, Keyboard, dan monitor standar
2. Kebutuhan Perangkat Lunak
 - a. OS Windows 7/8/8.1/10
 - b. Xampp v5.6.31
 - c. Protege v4.3
 - d. Server Apache Jena Fuseki