



ITS

Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

PENERAPAN TEKNOLOGI SEMANTIC WEB UNTUK APLIKASI ENSIKLOPEDIA TANAMAN OBAT

Zainul Alim

NRP 5211 100 020

Dosen Pembimbing :

Radityo Prasetyanto Wibowo, S.Kom, M.Kom.

Hatma Suryotrisongko, S.Kom, M.Eng.

JURUSAN SISTEM INFORMASI

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2015



ITS

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 141501

THE USE OF SEMANTIC WEB TECHNOLOGIES FOR ENCYCLOPEDIA OF MEDICINAL PLANTS APPLICATIONS

Zainul Alim
NRP 5211 100 020

Supervisor :
Prasetianto Wibowo, S.Kom, M.Kom.
Hatma Suryotrisongko, S.Kom, M.Eng.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

PENERAPAN TEKNOLOGI SEMANTIC WEB UNTUK APLIKASI ENSIKLOPEDIA TANAMAN OBAT

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memp peroleh Gelar Sarjana Komputer
pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

Zainul Alim

NRP. 5211 100 020

01-09-2015

KETUA JURUSAN SISTEM INFORMASI



Dr. Eng. FEBRIYAN SAMOPA, S.Kom., M.Kom.

NIP.19730219 199802 1 001

LEMBAR PERSETUJUAN
PENERAPAN TEKNOLOGI SEMANTIC WEB
UNTUK APLIKASI ENSIKLOPEDIA
TANAMAN OBAT

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Oleh :

Zainul Alim
NRP. 5211 100 020

Disetujui Tim Penguji : Tanggal Ujian : 13 Juli 2015
Periode Wisuda : September 2015

Radityo Prasetyanto W., S.Kom, M.Kom.

(Pembimbing I)

Hatma Suryotrisongko, S.Kom, M.Eng.

(Pembimbing II)

Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom

(Penguji I)

Nisfu Asrul Sani, S.Kom, M.Sc.

(Penguji II)

PENERAPAN TEKNOLOGI SEMANTIC WEB UNTUK APLIKASI ENSIKLOPEDIA TANAMAN OBAT

Nama Mahasiswa : : Zainul Alim
NRP : 5211 100 020
Jurusan : Sistem Informasi FTIf-ITS
Pembimbing 1 : Radityo Prasetyanto W., S.Kom, M.Kom.
Pembimbing 2 : Hatma Suryotrisongko, S.Kom, M.Eng.

ABSTRAK

Perkembangan teknologi informasi kian pesat dan pemanfaatan teknologi informasi sudah diterapkan di berbagai bidang. Ensiklopedia untuk tanaman obat sampai saat ini masih terbatas pada buku cetak dan e-book. Sudah ada website penyedia informasi mengenai tanaman obat serta khasiatnya, namun bentuknya masih berupa postingan dengan beberapa kategori, blog serta forum konsultasi. Salah satu kekurangannya adalah kesulitan untuk mencari informasi tertentu dan hasilnya kurang relevan dan keragaman informasi menyebabkan kendala pada pertukaran informasi dimana informasi akan sulit untuk dipertemukan serta pemanfaatan data yang ada masih belum bisa dimanfaatkan secara maksimal.

Berdasarkan permasalahan di atas, penulis merancang ontologi tanaman obat yang diimplementasikan dalam aplikasi web ensiklopedia tanaman obat. Dalam Tugas Akhir ini akan dibahas tentang pembuatan ontologi tanaman obat serta penerapannya dalam aplikasi ensiklopedia tanaman obat.

Kata Kunci: *tanaman obat, ensiklopedia, semantic web, ontology*

THE USE OF SEMANTIC WEB TECHNOLOGIES FOR ENCYCLOPEDIA OF MEDICINAL PLANTS APPLICATIONS

Student Name : Zainul Alim
NRP : 5211 100 020
Department : Sistem Informasi FTIf-ITS
Supervisor 1 : Radityo Prasetyanto W., S.Kom, M.Kom.
Supervisor 2 : Hatma Suryotrisongko, S.Kom, M.Eng.

ABSTRACT

Information technology has developed fast and it has been implemented and used in many fields. Until now, encyclopedia for medicinal plants is still in printed-books, and e-books. There are websites providing informations about medicinal plants and it's benefits, but it is in blog, or forum form. The lack of this situation is difficult to looking for information about information of medicinal plants and the result often is not relevant and the variety or diversity of informations makes inter-linked and exchange of data still difficult because there is no relation and no something that can connect between data, also that data can not be used maximally.

Base on problem above, it is required to develop ontology of medicinal plants and then to be implemented in encyclopedia of medicinal plants web applications. This Final Project will study about developing ontology of medicinal plants also it's use for encyclopedia of medicinal plants web applications.

Keywords: medicinal plants, encyclopedia, semantic web, ontology

KATA PENGANTAR

Segala puji dan syukur penulis haturkan ke hadirat Allah SWT yang telah memberikan kekuatan dan kehidupan untuk penulis sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul:

PENERAPAN TEKNOLOGI SEMANTIC WEB UNTUK APLIKASI ENSIKLOPEDIA TANAMAN OBAT

Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang telah meluangkan waktu, tenaga dan pikirannya bagi penulis untuk menyelesaikan tugas akhir ini. Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak-pihak di bawah ini, yaitu:

- Allah SWT yang telah memberi segala rahmat dan hidayah untuk dapat menyelesaikan tugas belajar selama di Sistem Informasi ITS dan telah memberikan kemudahan selama pengerjaan Tugas Akhir ini.
- Nabi Besar Muhammad SAW yang memberikan contoh semangat juang hingga akhir.
- Kedua orang tua serta keluarga penulis yang selalu memberikan dukungan dan kepercayaan. Terima kasih atas doa dan dukungannya yang tiada henti.
- Bapak Radityo Prasetyanto Wibowo, selaku dosen pembimbing I dan Bapak Hatma Suryotrisongko, selaku dosen pembimbing II yang telah meluangkan waktu dan pikiran di tengah kesibukan beliau untuk membimbing dan mengarahkan penulis dalam mengerjakan tugas akhir ini hingga selesai.
- Bigbanger famili yang telah mendukung, membantu, dan memberikan masukan kepada penulis selama

masa pengerjaan Tugas Akhir ini serta terima kasih telah memberikan hari-hari yang berwarna untuk penulis dari awal hingga akhir perkuliahan.

- Siti Nurlaila yang telah sabar dalam membantu, mengarahkan, menemani, dan menyemangati penulis selama pengerjaan Tugas Akhir ini.
- Anggota dan laboran laboratorium E-Bisnis yang tidak, bisa disebutkan satu persatu dan dan tak henti-hentinya memberikan dukungan kepada penulis.
- BASILISK, angkatan 2011 Jurusan Sistem Informasi ITS, yang selalu memberikan dukungannya, menjadi teman diskusi, dan membantu penulis dalam menjalani perkuliahan.
- Kepada seluruh angkatan di Jurusan Sistem Informasi yang telah banyak mengajarkan banyak hal kepada penulis.
- Serta seluruh pihak yang telah membantu penulis dalam mengerjakan tugas akhir ini yang tidak mungkin disebutkan satu per satu.

Penulis pun menyadari bahwa tugas akhir ini masih belum sempurna dengan segala kekurangan di dalamnya. Oleh karena itu penulis memohon maaf atas segala kekurangan dan kekeliruan yang ada di dalam tugas akhir ini. Penulis membuka pintu selebarlebarnya bagi pihak-pihak yang ingin memberikan kritik dan saran bagi penulis untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi seluruh pembaca.

Surabaya, 2015

Penulis

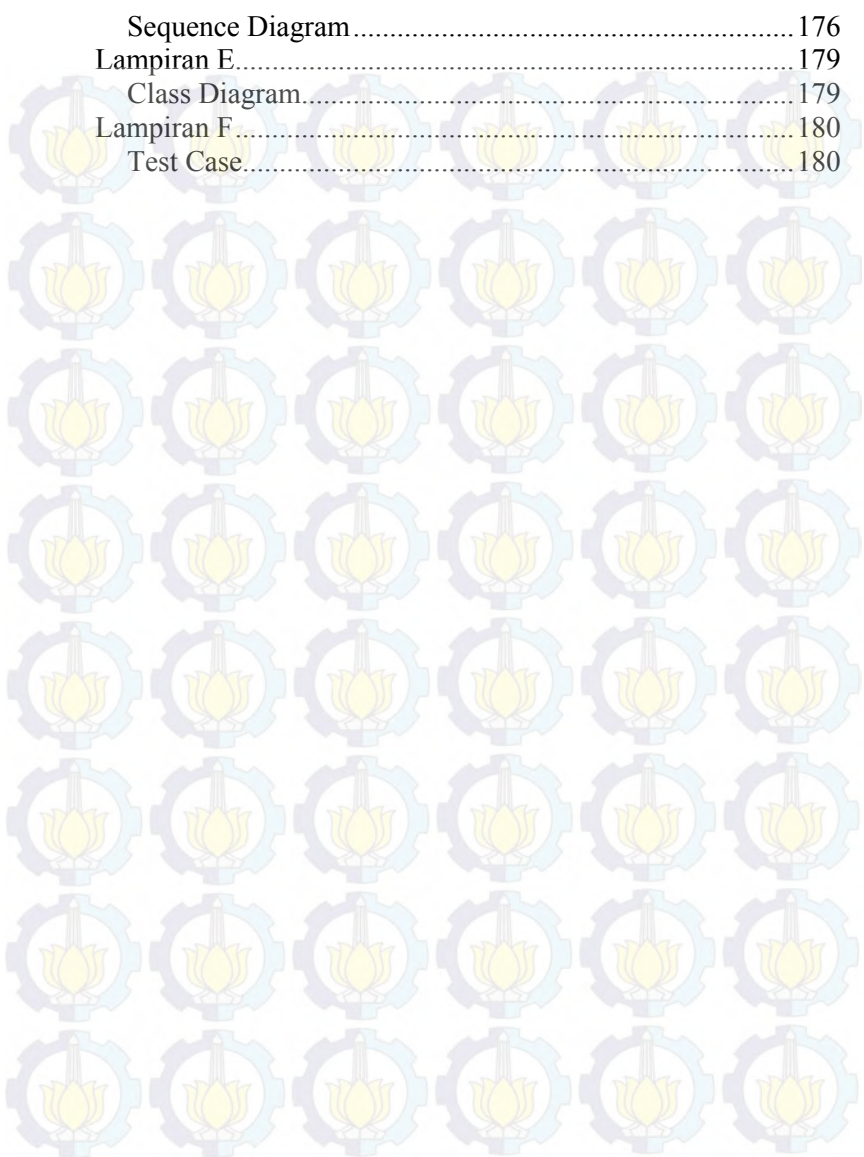
DAFTAR ISI

ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	3
1.3. Batasan Tugas Akhir	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Relevansi	5
1.7. Target Luaran	6
1.8. Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	9
2.1. Penelitian Sebelumnya	9
2.2. Dasar Teori	10
2.2.1. Tanaman Obat	10
2.2.2. Ensiklopedia	14
2.2.3. <i>World Wide Web</i>	16
2.2.4. <i>Semantic Web</i>	24
2.2.5. DBpedia	40
BAB III METODOLOGI PENELITIAN	41
3.1. Pengumpulan Data	41
3.2. Analisis Kebutuhan	42
3.3. Desain Aplikasi	43
3.4. Pembuatan Ontologi	44
3.5. Pembuatan Aplikasi	45
3.6. Pengujian Aplikasi	47
3.7. Penyusunan Buku Tugas Akhir	47
BAB IV ANALISIS KEBUTUHAN DAN DESAIN SISTEM	49
4.1. Deskripsi Umum Sistem	49
4.2. Analisis Kebutuhan	50

4.2.1.	Analisis Kebutuhan Pengguna Aplikasi.....	51
4.2.2.	Analisis Kebutuhan Pengguna Ontologi	51
4.2.3.	Fitur Aplikasi	51
4.2.4.	Arsitektur Sistem.....	52
4.3.	Domain Model	56
4.4.	Use Case Model	57
4.4.1.	Use Case Diagram.....	58
4.4.2.	Deskripsi Use Case	58
4.5.	GUI Prototype.....	59
4.6.	Robustness Diagram	60
4.7.	Sequence Diagram	60
4.8.	Class Diagram.....	60
4.9.	Test Case.....	61
BAB V PEMBUATAN ONTOLOGI		63
5.1.	Specification	63
5.2.	Knowledge Acquisition	63
5.3.	Conceptualisation.....	63
5.4.	Integration.....	68
5.4.1.	Pendefinisian Kelas.....	72
5.4.2.	Pendefinisian Properti	74
5.5.	Implementation	79
5.5.1.	Pendefinisian Ontologi.....	79
5.5.2.	Pembuatan Kelas.....	81
5.5.3.	Pembuatan Object Property.....	83
5.5.4.	Pembuatan Data Property.....	86
5.5.5.	Pendefinisian Relasi Antar Kelas dan Properti	88
5.5.6.	Pemasukan Data Tanaman Obat	89
5.6.	Evaluation	90
5.6.1.	Pemeriksaan Konsistensi	90
5.6.2.	Pemeriksaan Relasi	92
5.6.3.	Pemeriksaan DL-Query.....	93
5.6.4.	Pemeriksaan Query SPARQL.....	95
5.7.	Documentation.....	101
BAB VI IMPELEMENTASI DAN UJI COBA		103
6.1.	Lingkungan Implementasi	103

6.2.	Konfigurasi	104
6.2.1.	OpenRDF Sesame	104
6.2.2.	Pubby.....	106
6.2.3.	Direktori Aplikasi.....	108
6.2.4.	Konfigurasi Aplikasi	109
6.3.	Pembuatan Aplikasi	113
6.3.1.	Menu Utama.....	113
6.3.2.	Menu Daftar Tanaman Obat.....	115
6.3.3.	Menu Daftar Penyakit	119
6.3.4.	Menu Informasi Penyakit.....	122
6.4.	Uji Coba.....	124
6.4.1.	Fungsional	124
6.4.2.	Perbandingan.....	125
6.4.3.	Verifikasi Endpoint	129
6.4.4.	<i>Stress Testing</i>	130
6.5.	Pemanfaatan Ontologi.....	148
BAB VII HASIL DAN PEMBAHASAN		151
7.1.	Fungsional.....	151
7.2.	Perbandingan	152
7.3.	Verifikasi Endpoint.....	154
7.4.	<i>Stress Testing</i>	154
BAB VIII KESIMPULAN DAN SARAN		157
8.1.	Kesimpulan	157
8.2.	Saran	158
DAFTAR PUSTAKA.....		159
BIOGRAFI PENULIS.....		163
LAMPIRAN-LAMPIRAN		165
Lampiran A		165
Use Case Diagram		165
Use Case Skenario		166
Lampiran B		170
GUI Prototype.....		170
Lampiran C		173
Robustness Diagram		173
Lampiran D		176

Sequence Diagram.....	176
Lampiran E.....	179
Class Diagram.....	179
Lampiran F.....	180
Test Case.....	180



DAFTAR GAMBAR

Gambar 1. 1 Pohon Penelitian Laboratorium E-Bisnis	5
Gambar 2. 1 Akar Gingseng dan Kolesom.....	13
Gambar 2. 2 Perkembangan Web.....	17
Gambar 2. 3 Contoh kode Javascript.....	20
Gambar 2. 4 Layer <i>Web Semantic</i>	24
Gambar 2. 5 RDF Graph Mendeskripsikan Eric Miller	28
Gambar 2. 6 Contoh query SPARQL	31
Gambar 2. 7 Representasi query SPARQL ke dalam ajabar relasional	31
Gambar 2. 8 Contoh RDF Tuple	32
Gambar 2. 9 Contoh RDF Term/Triple	32
Gambar 2. 10 Contoh Relasi RDF	33
Gambar 2. 11 RDF Graph	33
Gambar 2. 12 Selection	34
Gambar 2. 13 Contoh selection pada SPARQL	34
Gambar 2. 14 Projection.....	34
Gambar 2. 15 Contoh projection pada SPARQL	35
Gambar 2. 16 Contoh tuple A dan B	35
Gambar 2. 17 Inner join	36
Gambar 2. 18 Left outer join	36
Gambar 2. 19 Contoh Inner join pada SPARQL	37
Gambar 2. 20 Contoh left outer join pada SPARQL	37
Gambar 2. 21 Tuple A dan B	37
Gambar 2. 22 A Union B.....	38
Gambar 2. 23 Contoh union pada SPARQL	38
Gambar 2. 24 Contoh Antarmuka Protege	39
Gambar 3. 1 Metodologi Penelitian	41
Gambar 3. 2 Gambaran Umum Cara Kerja Aplikasi	46
Gambar 4. 1 Arsitektur Sistem	53
Gambar 4. 2 Domain Model.....	57

Gambar 5. 1 Contoh URI Jahe pada id.dbpedia.org.....	64
Gambar 5. 2 Contoh Representasi Tanaman Obat	67
Gambar 5. 3 Contoh hubungan Jeruk Nipis dengan Batuk	68
Gambar 5. 4 Pemisalan kondisi berbeda	69
Gambar 5. 5 Gambaran ontologi sebelum penambahan kelas Recipe.....	70
Gambar 5. 6 Gambaran ontologi setelah penambahan kelas Recipe.....	71
Gambar 5. 7 Contoh penerapan dengan Jeruk Nipis	72
Gambar 5. 8 Kelas pada Ontologi Tanaman Obat.....	73
Gambar 5. 9 Pendefinisian Nama Ontologi.....	79
Gambar 5. 10 Memasukkan Prefix.....	80
Gambar 5. 11 Daftar Kelas pada Protege	82
Gambar 5. 12 Anotasi Kelas.....	82
Gambar 5. 13 Instance pada Kelas PlantPart.....	83
Gambar 5. 14 Memasukkan Object Property	85
Gambar 5. 15 Pendefinisian Invers	86
Gambar 5. 16 Anotasi Object Property	86
Gambar 5. 17 Memasukkan Data Property	87
Gambar 5. 18 Anotasi Data Property	88
Gambar 5. 19 Pendefinisian Relasi	89
Gambar 5. 20 Memasukkan Instance Tanaman Obat.....	90
Gambar 5. 21 Memilih dan menjalankan Reasoner HermiT	91
Gambar 5. 22 Proses me-load data.....	91
Gambar 5. 23 Hasil reasoning HermiT	92
Gambar 5. 24 Melihat relasi "Batuk"	93
Gambar 5. 25 Hasil visualisasi untuk menggambarkan hubungan "Batuk" dengan kelas atau objek yang lain	93
Gambar 5. 26 Menjalankan Reasoner	94
Gambar 5. 27 DL-Query untuk melihat tanaman obat untuk obat Batuk	94
Gambar 5. 28 Hasil tanaman obat untuk obat penyakit "Batuk".	95
Gambar 5. 29 Query SPARQL untuk menghitung jumlah tanaman obat.....	96

Gambar 5. 30 Jumlah Tanaman Obat	96
Gambar 5. 31 SPARQL query untuk mengecek properti	97
Gambar 5. 32 Hasil jumlah untuk tiap properti	97
Gambar 5. 33 Query untuk menghitung jumlah resep	98
Gambar 5. 34 Hasil jumlah resep	98
Gambar 5. 35 Query untuk melihat jumlah tanaman, penyakit, dan resep	99
Gambar 5. 36 Hasil jumlah tanaman, penyakit dan resep	99
Gambar 5. 37 Query untuk menghitung jumlah tanaman tiap kategori	100
Gambar 5. 38 Hasil jumlah tanaman obat tiap kategori	100
Gambar 5. 39 Query untuk melihat invers dari treatmentFor	101
Gambar 5. 40 Hasil invers dari treatmentFor	101
 Gambar 6. 1 Mengisi Informasi Repository	 104
Gambar 6. 2 Mengunggah File Ontologi *.owl	105
Gambar 6. 3 Menjalankan Query	105
Gambar 6. 4 Hasil Query	106
Gambar 6. 5 Kode config.ttl	107
Gambar 6. 6 Salah Satu Halaman untuk Instance Randu	108
Gambar 6. 7 Direktori Aplikasi	108
Gambar 6. 8 File library SPARQL PHP-Lib	109
Gambar 6. 9 Penggunaan library	110
Gambar 6. 10 Contoh Query SPARQL	110
Gambar 6. 11 Kelas pada model.php	111
Gambar 6. 12 Method pada Kelas Herb	112
Gambar 6. 13 Method pada Kelas Disease	112
Gambar 6. 14 Method pada Kelas Recipe	113
Gambar 6. 15 Fitur Pencarian	113
Gambar 6. 16 Memanggil Method findHerb	114
Gambar 6. 17 Potongan kode pada method findHerb	114
Gambar 6. 18 Hasil pencarian dengan kata kunci "wasir"	115
Gambar 6. 19 Indeks Abjad	116
Gambar 6. 20 Daftar Nama Tanaman Obat yang diawali dengan huruf "A"	116

Gambar 6. 21 Potongan kode pada method <code>getHerbByAlphabet</code>	117
Gambar 6. 22 Peringatan bahawa tidak ada tanaman obat	117
Gambar 6. 23 Informasi detail mengenai Akasia	118
Gambar 6. 24 Daftar Abjad	119
Gambar 6. 25 Daftar Penyakit yang diawali dengan huruf "A"	120
Gambar 6. 26 Peringatan bahawa tidak ada penyakit	120
Gambar 6. 27 Informasi detail mengenai Batuk	121
Gambar 6. 28 Potongan kode pada method <code>getDiseaseByURI</code>	122
Gambar 6. 29 Contoh informasi mengenai "Demam"	123
Gambar 6. 30 Potongan kode pada method <code>getDiseaseByName</code>	123
Gambar 6. 31 Pesan error ketika tidak ada informasi penyakit "Kulit gatal"	124
Gambar 6. 32 Hasil pencarian "jahe" pada aplikasi	126
Gambar 6. 33 Hasil pencarian "jahe" pada Google	126
Gambar 6. 34 Hasil pencarian "darah tinggi" pada aplikasi	127
Gambar 6. 35 Hasil pencarian "darah tinggi" pada Google	128
Gambar 6. 36 Hasil pencarian "obat herbal darah tinggi" pada Google	128
Gambar 6. 37 Hasil pencarian "obat herbal darah tinggi" pada aplikasi	129
Gambar 6. 38 Potongan kode query	129
Gambar 6. 39 Hasil untuk nama dan nama ilmiah dari 5 tanaman obat	130
Gambar 6. 40 Membuat Test Plan	131
Gambar 6. 41 Mengatur banyaknya pengguna	132
Gambar 6. 42 Membuat HTTP Request	132
Gambar 6. 43 Mengatur alamat endpoint	133
Gambar 6. 44 Parameter simple 'query'	133
Gambar 6. 45 Grafik untuk 10 pengguna	134
Gambar 6. 46 Gambar tabel analisis untuk 10 pengguna	134
Gambar 6. 47 Grafik untuk 50 pengguna	135
Gambar 6. 48 Gambar tabel analisis untuk 50 pengguna	135
Gambar 6. 49 Grafik untuk 100 pengguna	136

Gambar 6. 50 Gambar tabel analisis untuk 100 pengguna	136
Gambar 6. 51 Grafik untuk 200 pengguna	137
Gambar 6. 52 Gambar tabel analisis untuk 200 pengguna	137
Gambar 6. 53 Grafik untuk 500 pengguna	138
Gambar 6. 54 Gambar tabel analisis untuk 500 pengguna	138
Gambar 6. 55 Parameter advance 'query'	139
Gambar 6. 56 Grafik untuk 10 pengguna	140
Gambar 6. 57 Gambar tabel analisis untuk 10 pengguna	140
Gambar 6. 58 Grafik untuk 50 pengguna	141
Gambar 6. 59 Gambar tabel analisis untuk 50 pengguna	141
Gambar 6. 60 Grafik untuk 100 pengguna	142
Gambar 6. 61 Gambar tabel analisis untuk 100 pengguna	142
Gambar 6. 62 Grafik untuk 50 pengguna	143
Gambar 6. 63 Gambar tabel analisis untuk 200 pengguna	143
Gambar 6. 64 Membuat skenario	144
Gambar 6. 65 Grafik untuk 10 pengguna	144
Gambar 6. 66 Gambar tabel analisis untuk 10 pengguna	145
Gambar 6. 67 Grafik untuk 50 pengguna	145
Gambar 6. 68 Gambar tabel analisis untuk 50 pengguna	146
Gambar 6. 69 Grafik untuk 100 pengguna	146
Gambar 6. 70 Gambar tabel analisis untuk 100 pengguna	147
Gambar 6. 71 Grafik untuk 200 pengguna	147
Gambar 6. 72 Gambar tabel analisis untuk 200 pengguna	148
 Gambar 7. 1 Grafik perbandingan hasil stres test.....	 155
 Gambar A. 1 Use Case Diagram	 165
 Gambar B. 1 Halaman Menu Utama	 170
Gambar B. 2 Halaman Daftar Abjad Tanaman Obat	170
Gambar B. 3 Halaman Informasi Tanaman Obat	171
Gambar B. 4 Halaman Daftar Penyakit.....	171
Gambar B. 5 Halaman Informasi Penyakit.....	172

Gambar C. 1 Diagram Robustness Use Case Melihat Daftar Tanaman Obat	173
Gambar C. 2 Diagram Robustness Use Case Melihat Informasi Tanaman Obat	173
Gambar C. 3 Diagram Robustness Use Case Melihat Daftar Penyakit	174
Gambar C. 4 Diagram Robustness Use Case Melihat Informasi Penyakit	174
Gambar C. 5 Diagram Robustness Use Case Mencari Tanaman Obat	175
Gambar D. 1 Diagram Sequence Use Case Melihat Daftar Tanaman Obat	176
Gambar D. 2 Diagram Sequence Use Case Melihat Informasi Tanaman Obat	176
Gambar D. 3 Diagram Sequence Use Case Melihat Daftar Penyakit	177
Gambar D. 4 Diagram Sequence Use Case Melihat Informasi Penyakit	177
Gambar D. 5 Diagram Sequence Use Case Mencari Tanaman Obat	178
Gambar E. 1 Class Diagram Ensiklopedia Tanaman Obat	179

DAFTAR TABEL

Tabel 2. 1 Daftar Penelitian Sebelumnya	9
Tabel 4. 1 Daftar Fitur Aplikasi	52
Tabel 4. 2 Daftar Use Case.....	58
Tabel 4. 3 Daftar GUI Prototype	59
Tabel 4. 4 Relasi Antar Class dan Use Case	61
Tabel 4. 5 Contoh Daftar Test Case	61
Tabel 5. 1 Hubungan Tanaman Obat dengan Manfaatnya	65
Tabel 5. 2 Hubungan Bagian Tanaman Obat untuk Penyakit	66
Tabel 5. 3 Daftar Property	66
Tabel 5. 4 Penjelasan Kelas pada Ontologi Tanaman Obat	73
Tabel 5. 5 Daftar Object Property	75
Tabel 5. 6 Daftar Data Property	76
Tabel 5. 7 Daftar Prefix.....	79
Tabel 5. 8 Daftar Kelas dan Sub Kelas.....	81
Tabel 5. 9 Daftar Object Property	83
Tabel 5. 10 Daftar Data Property	87
Tabel 5. 11 Dokumentasi Kegiatan	102
Tabel 6. 1 Spesifikasi Komputer	103
Tabel 6. 2 Software Pendukung Lain yang Digunakan	103
Tabel 6. 3 Hasil <i>Test Case</i>	124
Tabel 7. 1 Perbandingan aplikasi ensiklopedia tanaman obat dengan	153
Tabel 7. 2 Perbandingan jumlah pengguna dengan rata-rata lamanya waktu response.....	154
Tabel A. 1 Skenario Melihat Daftar Tanaman Obat.....	166
Tabel A. 2 Skenario Melihat Informasi Tanaman Obat	166
Tabel A. 3 Skenario Melihat Daftar Penyakit	167
Tabel A. 4 Skenario Melihat Informasi Penyakit.....	167

Tabel A. 5 Skenario Mencari Tanaman Obat.....	168
---	-----

Tabel F. 1 Melihat Abjad Tanaman Obat.....	180
--	-----

Tabel F. 2 Test Case Melihat Daftar Tanaman Obat.....	181
---	-----

Tabel F. 3 Test Case Melihat Informasi Ilmiah Tanaman Obat.....	182
---	-----

Tabel F. 4 Test Case Melihat Daftar Penyakit Untuk Tiap Tanaman Obat.....	183
---	-----

Tabel F. 5 Test Case Melihat Resep Obat Untuk Tiap Penyakit.....	184
--	-----

Tabel F. 6 Test Case Melihat Abjad Penyakit.....	185
--	-----

Tabel F. 7 Test Case Melihat Daftar Penyakit.....	186
---	-----

Tabel F. 8 Test Case Melihat Abstrak Penyakit.....	187
--	-----

Tabel F. 9 Test Case Melihat Gambar Ilustrasi Penyakit.....	189
---	-----

Tabel F. 10 Test Case Melihat Daftar Tanaman Obat Untuk Tiap Penyakit.....	190
--	-----

Tabel F. 11 Test Case Mencari Tanaman Obat.....	191
---	-----

DAFTAR PUSTAKA

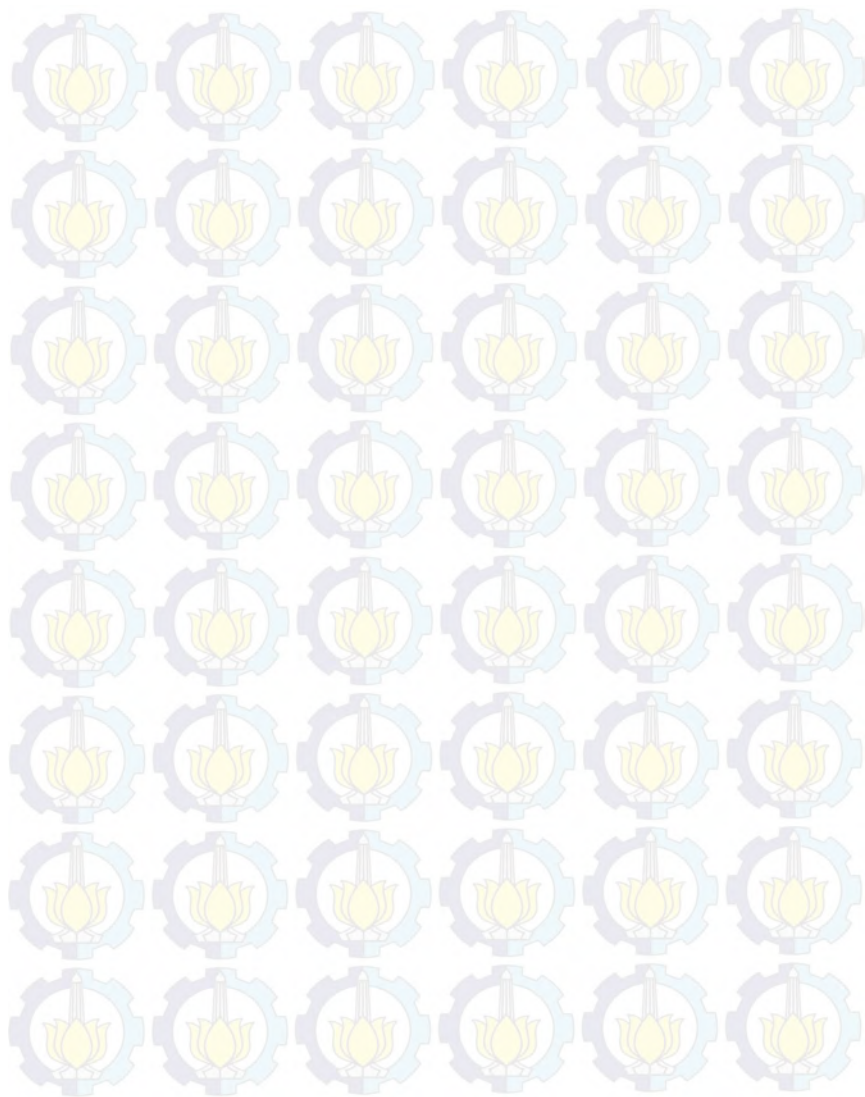
- [1] 3kbo, **Requirements of a Semantic Web Framework**, Building Semantic Web Applications, 31 Januari 2015, <<http://blog.3kbo.com/2008/10/10/requirements-of-asesemantic-web-framework>>.
- [2] Afif, N., **Implementasi Web Semantik Pada Pencarian Buku Perpustakaan UIN Alauddin Makassar**. Makassar, Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Alauddin Makassar, 2013.
- [3] Alim, Z., **Pembuatan Aplikasi Mobile Untuk Panduan Wisata Di Indonesia Berdasarkan Lokasi Dengan Menggunakan Semantic Web**, Manado, Konferensi Nasional Sistem Informasi 2014, 2014.
- [4] Ayuningtyas, N., **Implementasi Ontologi Web dan Aplikasi Web Semantik untuk Sistem Sitasi Jurnal Elektronik Indonesia**, Depok, Universitas Indonesia, 2009.
- [5] Balai Besar Penelitian dan Pengembangan Pascapanen Pertanian, **Teknologi Pascapanen Tanaman Obat**, Jakarta Selatan, Balai Besar Penelitian dan Pengembangan Pascapanen Pertanian, 2013.
- [6] Bengkel Internet PENS-ITS, **Modul Javascript**, Surabaya, PENS-ITS, 2006.
- [7] Bernard Renaldy Suteja, S. G., **Personalization Sistem E-Learning Berbasis Ontology**, Makara, 2010.
- [8] Berners-Lee, T. H., **The Semantic Web**, USA, Scientific American, 2001.
- [9] Dean Jones, T. B.-C., **Methodologies for Ontology Development**, Liverpool, Department of Computer Science, University of Liverpool, 2007

- [10] Dwiono, A., **Mesin Pencari Cerdas dengan Web Semantik**, Jurnal Generic, Vol . 8, No. 1, Maret 2013, 209-220, 2013.
- [11] Frisnanto, M., **Pembuatan Aplikasi Ensiklopedia Makanan Tradisional Berbasis Web Semantik**, Surabaya, Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember, 2010.
- [12] Kabar Toraja., **Apa itu ensiklopedia?** Hämtat från Kabar Toraja, 31 Januari 2015, <<http://kabartoraja.com/humaniora/pendidikan/1154-apa-ituensiklopedia>>.
- [13] Lestari, W., **Aplikasi Ensiklopedia Obat-Obat Herba**, Medan, Program Studi D-Iii Teknik Informatika Departemen Matematika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara, 2013.
- [14] Manoppo, M. P., **Pembuatan Aplikasi Ensiklopedia Tanaman Obat Tradisional Berbasis Sistem Operasi Android**, Surabaya: Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember, 2010.
- [15] Mirza Ilhami, S. A., **Pengembangan Website Portal Film Menggunakan Teknologi Web Semantik**, Medan, Study Program Of Informatics Engineering Sekolah Tinggi Manajemen Informatika Dan Komputer (STMIK) Mikroskil, 2011.
- [16] Nowack, B., **ARC RDF Classes for PHP**. Hämtat från GitHub ARC, 28 Januari 2015, <<https://github.com/semsol/arc2>>.
- [17] Petra., **Pengembangan Web Ontology**, Hämtat från Males Koding, 31 Januari 2015, <<https://maleskoding.wordpress.com/2009/08/26/pengembangan-web-ontology/>>
- [18] Th.Dwiati Wismarini, D. B., **Elektronik Ensiklopedi Tanaman Herba sebagai Bank Data Digital Tanaman**

Obat, Semarang, Program Studi Teknik Informatika, Universitas Stikubank, 2012.

- [19] Uzlina Rahmawati, E. S., **Pengembangan Repository Pengetahuan Berbasis Ontologi (Ontology-Driven Knowledge Repository) Untuk Tanaman Obat Indonesia**. Surabaya, Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS), 2012.
- [20] W3C, **Semantic Web**, W3C Org, 29 Juni 2015, <<http://w3c.org/standards/semanticweb/>>
- [21] Cyganiak, Richard, **A Relational Algebra for SPARQL**, Digital Media Systems Laboratory, HP Laboratories Bristol, 2005.

(halaman ini sengaja dikosongkan)

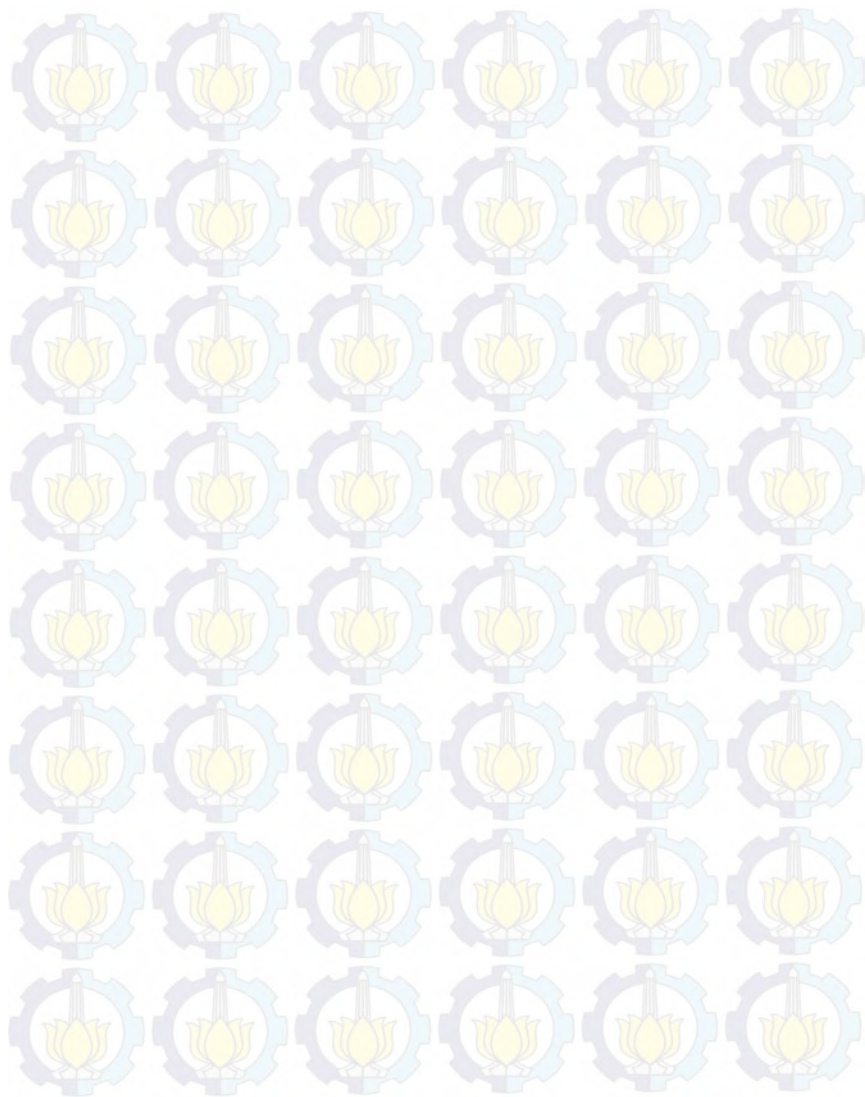


BIOGRAFI PENULIS



Penulis lahir di salah satu kabupaten yang ada di Jawa Timur yaitu Pasuruan pada 29 Agustus 1992. Penulis dibesarkan dalam lingkungan keluarga yang sederhana. Itu menjadi tantangan sendiri bagi penulis untuk selalu mengembangkan diri dan berusaha keras khususnya dalam masalah pendidikan. Penulis menempuh pendidikan formal mulai dari SD tahun 1999 sampai 2005 di SDN Wrti 02. Pada 2005 penulis melanjutkan pendidikan menengah pertama di MTs. Al-Yasini di bawah naungan Pondok Pesantren Terpadu Al-Yasini yang merupakan salah satu pondok pesantren moderen yang ada di Pasuruan sampai tahun 2008. Pada jenjang selanjutnya di pendidikan menengah atas, penulis mendapatkan beasiswa dari Kementrian Agama RI untuk bersekolah di MAN Insan Cendekia Serpong Kota Tangerang Selatan Propinsi Banten. Pada tahun 2011 penulis berkesempatan kuliah di Institut Teknologi Sepuluh Nopember Surabaya tepatnya di jurusan Sistem Informasi dengan beasiswa dari DIKTI selama masa perkuliahan. Selama kuliah, penulis pernah menjadi asisten dosen dan praktikum beberapa mata kuliah khususnya yang berhubungan dengan pemograman JAVA. Penulis juga pernah beberapa kali terlibat dalam proyek pengembangan perangkat lunak untuk beberapa yayasan di Surabaya. Penulis mempunyai hobi membaca dan mengunjungi beberapa wisata alam yang ada di Indonesia. Penulis menumpahkan bakat menulis di blog yang beralamat <http://alim92.site40.net/>.

(halaman ini sengaja dikosongkan)



BAB I

PENDAHULUAN

Pada bagian ini akan dijelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian yang mendasari penelitian tugas akhir, serta keterkaitan dengan roadmap Laboratorium E-Bisnis.

1.1. Latar Belakang Masalah

Pengobatan tradisional masih menjadi salah satu alternatif dalam metode pengobatan penyakit. Di Indonesia pun sudah lama menggunakan tanaman obat sebagai bahan untuk pengobatan tradisional tersebut. Bahkan di era modern pada saat ini dengan didukung oleh perkembangan teknologi pengobatan modern masih banyak masyarakat Indonesia yang menggunakan tanaman obat sebagai alternatif pengobatan.

Perkembangan teknologi informasi kian pesat. Pemanfaatan teknologi informasi dalam berbagai bidang pun sudah dapat dirasakan. Salah satu tren teknologi saat ini adalah *semantic web*. Teknologi ini memungkinkan informasi yang ada di web dapat dengan mudah dipahami oleh mesin. Penerapan *semantic web* sendiri sudah banyak dilakukan di berbagai bidang, contohnya adalah penerapan *semantic web* dalam sistem reparasi motor dan mobil. Tujuan utama dari *semantic web* adalah agar informasi-informasi lebih dimengerti oleh mesin. Salah satu konsep yang diusung oleh *semantic web* adalah dengan merepresentasikan pengetahuan dengan menggunakan ontologi. Dengan adanya ontologi ini dapat memudahkan mesin untuk memahami pengetahuan tersebut. Konsep yang diusung juga adalah *linked data*, dimana harapannya data yang ada di internet dapat saling terhubung sehingga memudahkan dalam pemanfaatan kembali data tersebut.

Ensiklopedia untuk tanaman obat sampai saat ini masih terbatas pada buku cetak dan e-book sehingga tidak semua orang dapat membaca serta memilikinya. Sampai saat ini juga, sudah ada

beberapa website penyedia informasi mengenai tanaman obat serta khasiatnya, namun bentuknya masih berupa postingan dengan beberapa kategori, blog serta forum konsultasi. Salah satu kekurangannya adalah pengunjung kadang kesulitan untuk mencari informasi tertentu yang spesifik. Informasi yang tersebar di internet (termasuk informasi tentang tanaman obat) memiliki keragaman dalam hal penyajiannya seperti format gambar, teks, suara atau video. Kemudahan dalam menggunakan internet membantu siapa saja yang membutuhkan informasi untuk mencari informasi yang dibutuhkan [19]. Namun seringkali informasi yang disajikan tidak sepenuhnya relevan dengan apa yang diinginkan oleh pengguna. Keragaman informasi kemudian menyebabkan kendala pada pertukaran informasi dimana informasi akan sulit untuk dipertemukan sehingga dalam melakukan pencarian, seseorang harus membuka banyak halaman informasi. Salah satu pendekatan yang memungkinkan untuk menjembatani masalah ini adalah pemanfaatan web semantic yang memanfaatkan ontologi. Pemanfaatan data yang ada masih belum bisa dimanfaatkan secara maksimal dikarenakan data yang ada tidak saling terhubung. Sehingga, diperlukan sebuah metode penyajian dan pencarian informasi yang dapat menyajikan informasi yang relevan dengan apa yang diinginkan pengguna dari seluruh informasi yang dimiliki dan juga untuk keberagaman data yang tersebar diperlukan suatu metode untuk menghubungkan data tersebut agar data yang ada dapat dibagikan serta digunakan kembali.

Oleh karena itu, penulis membangun suatu ontologi tanaman obat yang diimplementasikan dalam aplikasi web ensiklopedia tanaman obat. Dengan menggunakan teknologi *semantic web* (ontologi), semua informasi mengenai detail tanaman obat dan penyakit tersebut dapat diklasifikasikan sesuai dengan sifatnya masing-masing. Ensiklopedia tanaman obat ini akan berisikan penjelasan mengenai tanaman obat itu sendiri kemudian dihubungkan dengan penyakit yang dapat disembuhkan dengan

tanaman obat tertentu dengan resep untuk pembuatan obat. Kemudahan yang diberikan juga, pengguna dapat melakukan pencarian dengan kata kunci untuk mencari informasi yang lebih spesifik. Harapannya, aplikasi web ensiklopedia tanaman obat ini dapat menjadi acuan bagi masyarakat untuk menambah wawasan sehingga memotivasi masyarakat untuk dapat menggunakan tanaman obat dan dapat melestarikan tanaman obat tersebut. Serta dari hasil ontologi yang sudah dibuat tersebut nantinya dibuatkan endpoint sehingga ontologi tanaman obat dapat dibagikan serta digunakan kembali oleh pihak yang membutuhkannya tanpa perlu membuat ontologi tanaman obat baru.

1.2. Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas dapat dirumuskan sebagai berikut :

1. Bagaimana langkah-langkah dalam membuat ontologi tanaman obat?
2. Bagaimana cara membangun SPARQL endpoint untuk ontologi tanaman obat?
3. Bagaimana langkah-langkah dalam membuat aplikasi web ensiklopedia tanaman obat?
4. Bagaimana cara membangun aplikasi web yang memanfaatkan ontologi tersebut sebagai basis data?
5. Bagaimana menghubungkan informasi mengenai tanaman obat dan penyakit tersebut dengan sumber data yang sudah ada?

1.3. Batasan Tugas Akhir

Dalam pengerjaan tugas akhir ini, ada beberapa batasan masalah yang harus diperhatikan, yaitu sebagai berikut :

1. Aplikasi web ensiklopedia tanaman obat menyediakan informasi mengenai tanaman obat itu sendiri beserta keterkaitannya dengan penyakit.
2. Fitur pencarian dapat melakukan pencarian mengenai tanaman obat dan penyakit.

3. Sumber informasi detail mengenai tanaman obat dan penyakit dihasilkan dari DBpedia.org yang menyediakan SPARQL endpoint untuk melakukan query terhadap dataset DBPedia.
4. Informasi yang digunakan pada aplikasi ini adalah dari beberapa sumber buku ensiklopedia mengenai tanaman obat, buku-buku tanaman herbal.
5. Penerapan *semantic web* pada aplikasi ini pada penggunaan ontologi tanaman obat yang digunakan oleh aplikasi sebagai basis pengetahuan.
6. Ontologi tanaman obat dapat diakses melalui endpoint yang disediakan dengan SPARQL query SELECT.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijelaskan sebelumnya, maka tujuan yang akan dicapai dalam penelitian ini adalah:

1. Membuat ontologi tanaman obat dan membangun endpoint untuk ontologi tanaman obat.
2. Menghasilkan aplikasi web ensiklopedia tanaman obat.
3. Memberikan salah satu sumber tentang tanaman obat dan penyakit dengan lengkap.

1.5. Manfaat Penelitian

Tugas akhir ini diharapkan dapat memberikan manfaat antara lain:

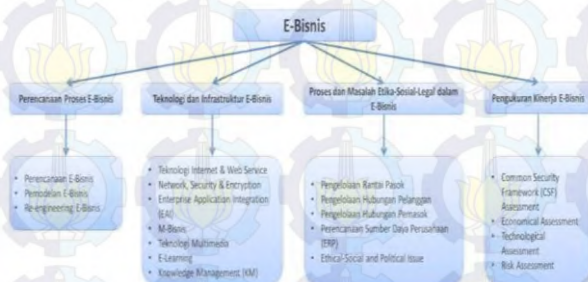
1. Sebagai referensi bagi seseorang yang ingin memanfaatkan tanaman obat sebagai salah satu alternatif pengobatan penyakit.
2. Memudahkan pengguna dalam menemukan informasi detail mengenai tanaman obat dan penyakit.
3. Memudahkan pengguna dalam pencarian semua informasi terkait dengan tanaman obat serta penyakit yang diinginkan, tanpa harus dirumitkan dengan pencarian menggunakan mesin pencari serta sumber lain.

4. Ontologi tanaman obat dapat diakses melalui endpoint yang disediakan dan dapat digunakan kembali oleh pihak yang membutuhkan.

1.6. Relevansi

Pembuatan dari tugas akhir ini berkaitan dengan Road Map Laboratorium E-Bisnis. Hal ini berkaitan dengan perkembangan penelitian yang dilakukan laboratorium E-bisnis. Rencana penelitian laboratorium E-Bisnis digambarkan dalam pohon penelitian laboratorium E-Bisnis. Pohon penelitian ini memiliki empat bahasan utama, yang diantaranya adalah Perencanaan Proses Bisnis, Teknologi dan Infrastruktur E-Bisnis, Proses dan masalah etika social-legal E-bisnis, dan Pengukuran Kinerja E-bisnis. Topik pada tugas akhir ini mengenai pembuatan aplikasi web ensiklopedia tanaman obat, dimana pada pohon penelitian Laboratorium E-Bisnis masalah tersebut terletak pada teknologi dan infrastruktur E-Bisnis khusus pada cabang Teknologi Internet dan Web Services. Diagram pohon penelitian laboratorium E-Bisnis yang sedang dikembangkan ditunjukkan oleh gambar 1.1 berikut ini.

Pohon Penelitian Laboratorium E-Bisnis



Gambar 1. 1 Pohon Penelitian Laboratorium E-Bisnis

1.7. Target Luaran

Adapun target luaran dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Ontologi tanaman obat dan penyakit yang dapat digunakan lagi nanti jika diperlukan serta endpoint untuk mengakses ontologi tersebut.
2. Aplikasi web ensiklopedia untuk referensi masyarakat mengenai tanaman obat.
3. Dokumentasi berupa buku Tugas Akhir dan Jurnal Ilmiah.

1.8. Sistematika Penulisan

Dalam penulisan pada buku tugas akhir ini, terdapat beberapa sistematika penulisan yang diterapkan. Berikut penjelasan dari tiap-tiap bab yang terdapat pada buku tugas akhir ini:

- **BAB I: PENDAHULUAN**
Berisi uraian mengenai inisiasi penelitian yang akan dilakukan, yaitu latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, relevansi, target luaran, dan sistematika penulisan.
- **BAB II: TINJAUAN PUSTAKA**
Pada bagian ini berisi penjelasan mengenai pustaka dan definisi dari setiap hal yang berkaitan dengan penelitian yang dapat dijadikan sebagai sumber referensi. Teori yang dipaparkan dalam bab ini, antara lain mengenai: tanaman obat, ensiklopedia, web beserta komponen-komponen di dalamnya, *semantic web* yang meliputi: RDF, OWL, SPARQL, serta tool yang digunakan dalam membangun web semantik.
- **BAB III: METODOLOGI**
Pada bagian ini dijelaskan mengenai pembuatan ontologi tanaman obat, sekaligus mengikuti dari metode pembuatan ontologi yaitu Methontology.
- **BAB IV: ANALISIS DAN DESAIN SISTEM**

Bab ini menjelaskan hal-hal terkait kebutuhan dan desain sistem pada aplikasi yang dibuat. Analisis kebutuhan sistem didasarkan pada aktivitas, user, fungsional, dan arsitektur. Desain sistem divisualisasikan dengan konsep Unified Modelling Language (UML) dan menggunakan model pengembangan ICONIX.

- **BAB V: PEMBUATAN ONTOLOGI**

Bab ini berisi hasil implementasi dari aplikasi yang telah dibangun beserta hasil pengujian aplikasi menggunakan system testing dan stress testing.

- **BAB VI: IMPLEMENTASI DAN UJI COBA**

Bab ini berisi hasil implementasi dari aplikasi yang telah dibangun beserta hasil pengujian aplikasi.

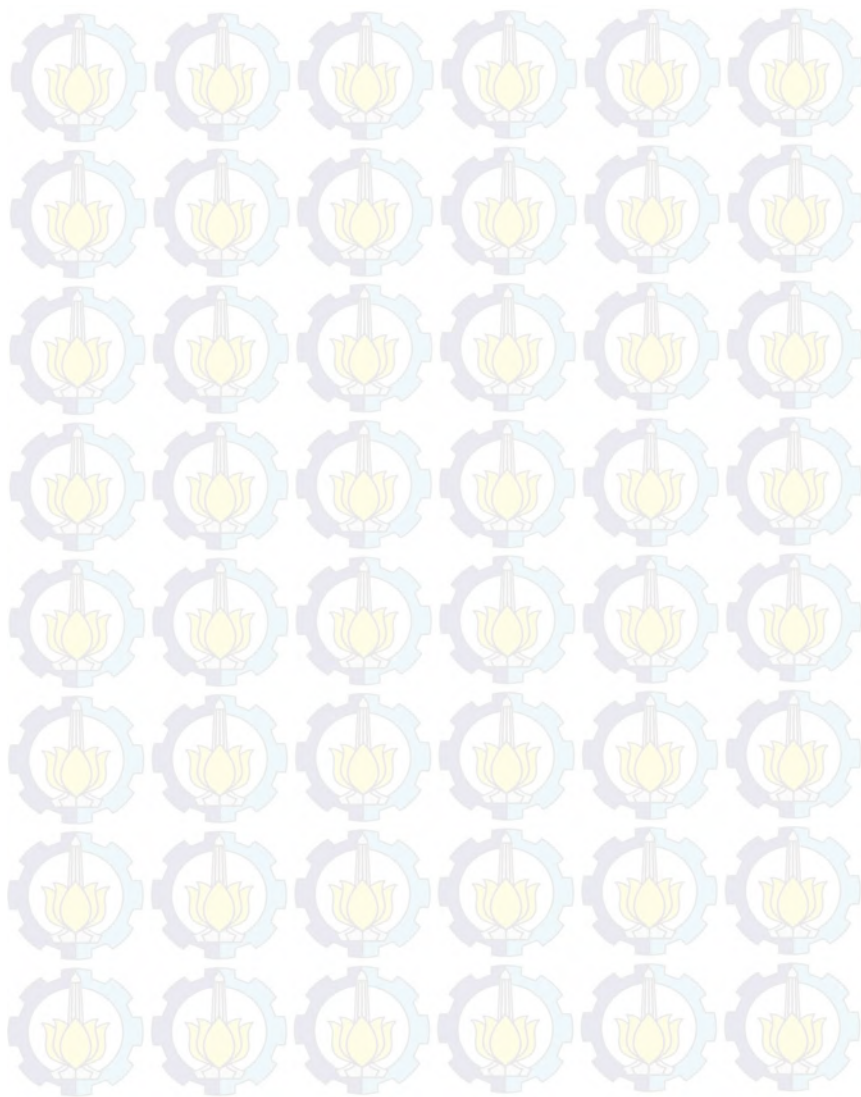
- **BAB VII: HASIL DAN PEMBAHASAN**

Bab ini berisi tentang pembahasan dari hasil yang didapatkan setelah dilakukan uji coba.

- **BAB VIII: KESIMPULAN DAN SARAN**

Berisi tentang simpulan dari keseluruhan tugas akhir dan saran maupun rekomendasi terhadap penelitian selanjutnya.

(halaman ini sengaja dikosongkan)



BAB II

TINJAUAN PUSTAKA

Pada bagian ini menjelaskan mengenai pustaka dan definisi dari setiap hal yang berkaitan dengan penelitian yang dapat dijadikan sebagai sumber referensi dalam pembuatan tugas akhir ini. Teori yang dipaparkan dalam bab ini, antara lain mengenai: tanaman obat, ensiklopedia, web beserta komponen-komponen di dalamnya, *semantic web* yang meliputi: RDF, OWL, SPARQL, serta tool yang digunakan dalam membangun web semantic, dan konsep-konsep lain yang berkaitan dengan pembuatan tugas akhir.

2.1. Penelitian Sebelumnya

Dalam mengerjakan tugas akhir ini terdapat penelitian terkait yang digunakan, berikut informasi singkat mengenai penelitian tersebut dijelaskan pada Tabel 2.1.

Tabel 2. 1 Daftar Penelitian Sebelumnya

Judul Penelitian	Tujuan Penelitian	Peneliti
Pembuatan Aplikasi Ensiklopedia Tanaman Obat Tradisional Berbasis Sistem Operasi Android.	Menjelaskan mengenai pembuatan aplikasi ensiklopedia dengan basis sistem operasi Android dan menggunakan basis data sederhana.	(Manoppo, M. P, 2010)
Elektronik Ensiklopedi Tanaman Herba sebagai Bank Data Digital Tanaman Obat.	Membahas mengenai cara pembuatan ensiklopedia sederhana berbentuk web dengan fitur pengurutan nama tanaman obat berdasarkan abjad.	(Th. Dwiati Wismarini, D. B, 2012)

Pengembangan Repository Pengetahuan Berbasis Ontologi (Ontology-Driven Knowledge Repository) Untuk Tanaman Obat Indonesia	Menjelaskan mengenai cara membuat ontologi tanaman obat dan melakukan query sederhana untuk mengambil informasi dari ontologi tersebut.	(Uzlina Rahmawati, E. S, 2012)
Pembuatan Aplikasi Ensiklopedia Makanan Tradisional Berbasis Web Semantik.	Membuat suatu ensiklopedi makanan tradisional. Aplikasi ensiklopedia dapat menampilkan daftar makanan tradisional serta menu pencarian yang telah menggunakan pencarian konteks, sehingga pengguna dapat melakukan pencarian yang lebih kompleks dan spesifik.	(Frisnanto, M, 2010)

2.2. Dasar Teori

2.2.1. Tanaman Obat

Tanaman obat yang berkembang di Indonesia sangat melimpah tetapi pemanfaatannya masih terbatas dikonsumsi secara segar, sehingga dibutuhkan teknologi pengolahan untuk dapat memaksimalkan pemanfaatannya. Pemanfaatan yang maksimal dari berbagai tanaman obat ini masih dirasa kurang beredar di masyarakat. Menurut Depkes RI, definisi tanaman obat sebagaimana tercantum dalam SK Menkes No.149/SK/Menkes/IV/1978, yaitu tanaman atau bagian

tanaman yang digunakan sebagai Obat Tradisional atau Jamu, tanaman atau bagian tanaman yang digunakan sebagai formula bahan baku obat, tanaman atau bagian tanaman yang diekstraksi, dan ekstraksi tersebut digunakan sebagai obat. Di Indonesia, dikenal lebih dari 20.000 jenis tumbuhan obat. Namun, baru 1.000 jenis tanaman telah terdata dan baru sekitar 300 jenis yang sudah dimanfaatkan untuk pengobatan tradisional [13].

2.2.1.1. Obat Herbal

Obat herbal adalah obat yang berasal dari tumbuhan yang diproses atau diekstrak sedemikian rupa sehingga menjadi serbuk, pil atau cairan yang dalam prosesnya tidak menggunakan zat kimia. Seperti yang di ketahui obat herbal dapat menyembuhkan penyakit dengan efek samping yang minim karena dibuat dari bahan-bahan yang alami, tidak seperti obat-obat sintetis yang dapat memberikan efek samping baik secara langsung maupun setelah waktu yang lama. Istilah Herbal biasanya dikaitkan dengan tumbuh-tumbuhan yang tidak berkayu atau tanaman yang bersifat perdu. Dalam dunia pengobatan, istilah herbal memiliki makna yang lebih luas, yaitu segala jenis tumbuhan dan seluruh bagianbagiannya yang yang mengandung satu atau lebih bahan aktif yang dapat dipakai sebagai obat (therapeutic). Misalnya mengkudu hutan (*Morinda citrifolia*) yang mengandung Morindin, bahan aktif anti kanker, Pegagan (*Centela Asiatica*) yang mengandung Asiaticoside yang berguna untuk masalah kulit dan meningkatkan IQ. Konsep Pengobatan Herbal sangat berbeda dengan konsep pengobatan Modern yang biasanya menggunakan Kimia Sintetis sebagai obat. Misalnya dalam pengobatan kimia sintetis penyebab penyakit adalah virus, bakteri, dan pathogen (mikro organisme pembawa penyakit) sedangkan dalam pengobatan herbal, penyebab penyakit adalah lemahnya sistem imun [13].

2.2.1.2. Khasiat dan Kegunaan

Khasiat merupakan suatu indikasi bahwa tanaman tersebut mempunyai kemampuan untuk dapat digunakan sebagai obat. Hal ini disebabkan oleh kandungan metabolit sekunder atau senyawa aktif yang memiliki daya kerja dalam pengobatan dari setiap tanaman. Khasiat dari suatu tanaman dapat diketahui setelah melalui proses uji manfaat atau praklinik dengan menggunakan hewan coba. Sebagai bahan baku yang digunakan untuk uji coba bisa berupa simplisia, sediaan galenik dan ekstrak yang telah memenuhi persyaratan minimal serta dapat menjamin keseragaman komponen aktif, keamanan dan kegunaannya. Sebagai contoh, uji anti hipertensi terhadap hewan coba dari ekstrak kasar daun belimbing wuluh dan yang telah di murnikan ternyata menunjukkan perbedaan dalam penurunan tekanan darah pada hewan coba. Ekstrak yang telah dimurnikan ternyata mempunyai efek penurunan tekanan darah lebih tinggi dibandingkan ekstrak kasar. Untuk durasi penurunan tekanan darah, ekstrak yang telah dimurnikan mempunyai waktu lebih lama dibandingkan ekstrak kasar. Hal ini dikarenakan, sebelum di murnikan ekstrak masih mengandung lilin, gula, gum, lemak sehingga akan mempengaruhi kandungan zat berkhasiatnya jadi rendah. Bila telah dimurnikan, senyawa yang tidak diinginkan telah terbuang, berarti kandungan zat berkhasiatnya menjadi lebih tinggi. Untuk obat hipertensi akan lebih baik bila mempunyai durasi penurunan tekanan darah yang tidak terlalu cepat.

Tanaman ginseng, sering digunakan dalam pengobatan tradisional terutama di negara China dan Korea (contoh pada Gambar 2.1). Khasiat dari ginseng dapat membantu proses pencernaan dan meningkatkan selera makan, obat diare, efektif mengobati asma dan gangguan pernapasan, merawat diabetes, rematik, bisul, dapat meminimalkan risiko terkena kanker, mengurangi stres fisik dan mental serta membantu memulihkan stamina, membantu meningkatkan sistem kekebalan tubuh,

efektif mengurangi kadar kolesterol jahat, dan memperlancar peredaran darah.



Gambar 2. 1 Akar Ginseng dan Kolesom

Di Indonesia terdapat tumbuhan yang memiliki khasiat yang hampir sama dengan ginseng yaitu *Talinum paniculatum* Gaertn atau dalam bahasa daerah disebut dengan som jawa / kolesom. Kajian mengenai khasiat dan kegunaanya telah dilakukan untuk menjadikan kolesom sebagai ginseng Indonesia. Beberapa uji farmakologis juga telah dilakukan [5].

2.2.1.3. Kandungan Kimia

Tanaman obat mengandung berbagai jenis senyawa kimia yang bisa berfungsi untuk mengobati berbagai macam penyakit dan juga berbagai macam jenis enzim. Enzim-enzim tertentu yang terdapat dalam tanaman harus di non aktifkan, bila masih bekerja maka senyawa kimia yang terdapat dalam tanaman akan berubah menjadi senyawa lain yang tidak mempunyai efek terapi. Didalam simplisia terdapat kandungan senyawa kimia, baik yang memiliki efek terapi yaitu senyawa aktif maupun yang tidak mempunyai efek terapi yaitu zat ballast seperti karbohidrat, lemak, protein, klorofil, resin dan tannin. Walaupun tidak memiliki efek terapi, akan tetapi zat ballast memiliki pengaruh yang cukup besar pada ekstraksi kandungan zat aktif.

Tanaman obat mengandung berbagai jenis senyawa kimia yang bisa berfungsi untuk mengobati berbagai macam penyakit dan juga berbagai macam jenis enzim. Enzim-enzim tertentu yang

terdapat dalam tanaman harus di non aktifkan, bila masih bekerja maka senyawa kimia yang terdapat dalam tanaman akan berubah menjadi senyawa lain yang tidak mempunyai efek terapi. Didalam simplisia terdapat kandungan senyawa kimia, baik yang memiliki efek terapi yaitu senyawa aktif maupun yang tidak mempunyai efek terapi yaitu zat ballast seperti karbohidrat, lemak, protein, khlorofil, resin dan tannin. Walaupun tidak memiliki efek terapi, akan tetapi zat ballast memiliki pengaruh yang cukup besar pada ekstraksi kandungan zat aktif [5].

2.2.2. Ensiklopedia

Kata “Ensiklopedia” berasal dari bahasa Yunani enkyklios paideia (ἐγκύκλιος παιδεία) yang berarti sebuah lingkaran atau pengajaran yang lengkap. Maksud dari ensiklopedia di sini yaitu pendidikan paripurna yang mencakup semua lingkaran ilmu pengetahuan [18]. Ensiklopedia adalah sejumlah tulisan yang berisi penjelasan yang menyimpan informasi secara komprehensif dan cepat dipahami serta dimengerti mengenai keseluruhan cabang ilmu pengetahuan atau khusus dalam satu cabang ilmu pengetahuan tertentu yang tersusun dalam bagian artikel-artikel dengan satu topik bahasan pada tiap-tiap artikel yang disusun berdasarkan abjad, kategori atau volume terbitan dan pada umumnya tercetak dalam bentuk rangkaian buku yang tergantung pada jumlah bahan yang disertakan [11]. Sedangkan menurut Kamus Besar Bahasa Indonesia ensiklopedia adalah “buku (atau serangkaian buku) yang menghimpun keterangan atau uraian tentang berbagai hal dalam bidang seni dan ilmu pengetahuan, yang disusun menurut abjad atau menurut lingkungan ilmu”. Seiring berkembangnya dunia ilmu pengetahuan dan teknologi, masyarakat semakin memudahkan untuk memperoleh informasi. Termasuk perubahan bentuk media ensiklopedia, apabila dulu untuk melihat sebuah informasi secara detail perlu pergi ke toko buku atau perpustakaan untuk mendapatkan buku

ensiklopedia, maka saat ini sudah cukup banyak situs web yang menyediakan layanan berupa ensiklopedia online yang dapat diakses dengan mudah kapanpun dimanapun. Saat ini sudah banyak web ensiklopedia yang ada di dunia. Web Ensiklopedia yang banyak diketahui orang adalah Wikipedia. Wikipedia adalah salah satu web ensiklopedia yang dibangun oleh para sukarelawan dan disediakan secara gratis oleh Wikimedia Foundation. Wikipedia memiliki modul-modul ensiklopedia yang lengkap, mulai dari budaya, geografi sejarah dan lain sebagainya. Untuk mencari keterkaitan antar informasi yang terkandung dalam suatu konten, Wikipedia memberikan semacam tautan dalam konten tersebut yang mengaitkan dengan artikel lain. Pemberian tautan ini bertujuan agar pengguna dapat mencari informasi yang terkait dengan menekan tautan tersebut. Dari hasil pengamatan yang dilakukan, tautan yang diberikan tersebut terkadang tidak konsisten [11].

Pada umumnya Ensiklopedia terbagi dalam 2 (dua) kategori, yakni: Ensiklopedia Umum (General Encyclopaedia), yaitu Ensiklopedia yang memuat secara umum semua disiplin ilmu didalamnya. Misalnya “Encyclopaedia Britannica edisi 2010”, 32 Jilid memuat semua disiplin ilmu dan semua peristiwa penting di dunia (dari A sampai Z). Ensiklopedia Khusus (Specialist Encyclopaedia), yaitu Ensiklopedia yang memuat disiplin ilmu/cabang ilmu tertentu atau bidang tertentu.

Misalnya “Ensiklopedia Geografi”, “Ensiklopedia Sejarah dan Budaya”, “Ensiklopedia IPA” (bidang IPA), “Ensiklopedia IPTEK” (bidang IPA), “Ensiklopedia Sains dan Teknologi” (bidang IPA), “Ensiklopedia Jakarta” (Tematis atau Historis Kronologis). Ensiklopedia memiliki ciri-ciri, diantaranya:

1. Adanya artikel/topik, sub topik.
2. Adanya definisi artikel/topik dan diikuti penjelasan umum.
3. Adanya rujuk silang (cross reference) atau further more, see also.

4. Adanya paragraf, ilustrasi, gambar, grafik, table time line.
5. Tersusun dengan baik, contohnya disajikan secara sistematis alfabetis (A – Z), atau tematis, historis, kronologis.
6. Adanya indeks [12].

2.2.3. World Wide Web

World Wide Web, yang sering disingkat WWW, merupakan sebuah sistem dokumen hypertext yang saling berhubungan dan tergabung ke dalam internet. Dokumen hypertext biasanya berupa tulisan (informasi), gambar, video, maupun beragam konten multimedia yang ditampilkan dengan menggunakan web browser. Dokumen-dokumen hypertext tersebut saling terhubung melalui penghubung yang dikenal dengan hyperlink. Kendatipun banyak dijumpai pada internet, WWW juga dapat dijalankan di dalam Intranet. Pertama kali berjalan pada tahun 1990, WWW yang merupakan sistem client-server di mana client dan server berkomunikasi menggunakan penanda unik tersebut yaitu Uniform Sumber daya Locator (URL), Uniform Sumber daya Identifier (URI), Hypertext Markup Language (HTML), dan Hypertext Transfer Protocol (HTTP).

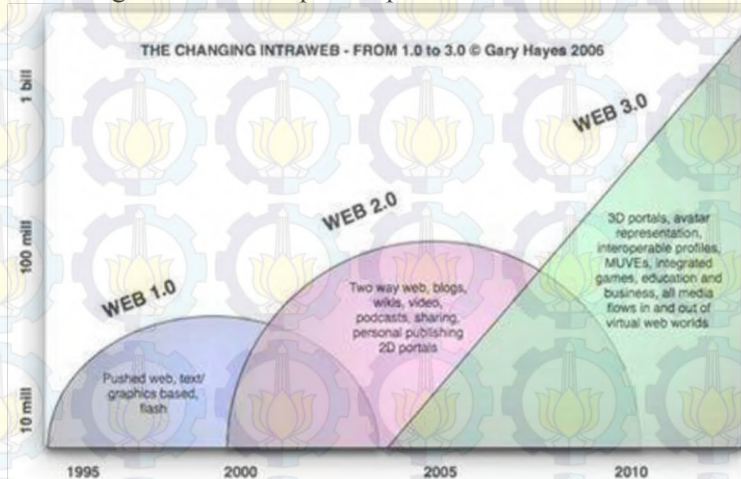
Keunggulan utama WWW pada saat pertama kali diluncurkan yaitu penggunaan hyperlink yang unidireksi, yang memungkinkan bagi seseorang untuk berhubungan dengan sumber daya di tempat lain tanpa perlu intervensi pemilik sumber daya. Hal ini mengurangi kerumitan implementasi web server dan browser secara signifikan, sehingga WWW dengan mudah dan cepat diadopsi oleh banyak pihak. Popularitas WWW terutama terlihat dari bagaimana website dibangun di seluruh dunia, dan bahkan sering WWW disamakan dengan Internet.

Dalam membangun WWW, terdapat banyak teknologi yang terlibat dari mulai persiapan server sampai ke penyajian konten

kepada pengguna. Beberapa teknologi yang memiliki peran cukup penting dalam membangun WWW yaitu TCP/IP, URI, DNS, Hypertext dan Hypermedia, HTML, HTTP serta AJAX [15].

2.2.3.1. Perkembangan Web

Seperti yang kita ketahui, web kini sudah mengalami perkembangan. Sejalan perkembangan web, semakin banyak pula fitur yang ditawarkan dan semakin bertambah juga kegunaannya. Pengembangan web ini sampai sekarang terus berlanjut dan melahirkan berbagai generasi web. Perkembangan web digambarkan seperti pada Gambar 2.2 berikut.



Gambar 2. 2 Perkembangan Web

Web 1.0 merupakan web yang statis dan hanya bisa dibaca. Ini merupakan versi web paling awal yaitu dimana web hanya memiliki satu arah, “read-only”. Oleh karena itu, versi web ini tidak dapat penggunai fasilitas untuk menulis komentar atau sebagainya. Kita hanya bisa membaca tanpa bisa

menyampaikan respon kembali. Elemen-elemen yang biasa digunakan dalam web 1.0 adalah:

- Halaman statis
- Framesets
- Tombol GIF

Contoh yang paling mudah dalam web 1.0 adalah situs pribadi. Berbeda dengan blog, situs pribadi hanya berpusat pada satu orang dan kita tidak bisa menggunakan tanggapan didalamnya.

Web 2.0 sudah mengalami perkembangan dari versi 1.0. Pada versi 2.0 kali ini kita sudah bisa meninggalkan komentar pada web tersebut. Sehingga bisa kita sebut sebagai “read-write”. Versi ini juga sudah menggunakan Rich-Internet Applications seperti AJAX, Flash, Flex, Java, Silverlight, dan beberapa lainnya. Aplikasi tersebut digunakan untuk mengupdate tanpa harus merefresh halaman. Secara fungsional, web 2.0 dibuat berdasarkan rancangan web server yang telah tersedia, akan tetapi menitik beratkan pada penggunaan software. Dalam web 2.0 juga terdapat beberapa sindikasi seperti RSS (Really Simple Syndication), Atom, dan masih banyak lagi yang disimpan dalam format XML (eXtendend Markup Language).

Web versi 3.0 merupakan yang paling terbaru sekarang ini karena menawarkan beberapa fitur baru seperti video streaming, audio streaming, dan masih banyak lagi. Web 3.0 juga biasa disebut “executable web” karena kemampuannya mengeksekusi suatu program. Jika dikaitkan dengan beberapa trilogi perkembangan web, maka web 3.0 adalah web yang bersifat “read-write-execute”. Belum banyak yang bisa diketahui dari web 3.0 karena versi ini hingga saat ini masih dalam tahap pengembangan yang lebih luas [15].

2.2.3.2. Web Design

Web desain melibatkan standar untuk membangun dan merender halaman web, termasuk HTML, CSS, SVG Ajax dan teknologi lainnya untuk aplikasi web.

2.2.3.3. HTML & CSS

HTML (Hypertext Markup Language) dan CSS (Cascading Style Sheets) adalah dua teknologi inti untuk membangun halaman web. HTML menyediakan struktur dari halaman sedangkan CSS digunakan untuk memvisualisasikan halaman web. Dengan menggunakan graphic dan scripting, HTML dan CSS adalah dasar dalam aplikasi web. HTML adalah bahasa untuk mendeskripsikan struktur dari halaman web. HTML digunakan untuk:

- Mempublish dokumen secara online dengan *heading, text, table, list, photo*, dan lain-lain.
- Menerima informasi secara online melalui hypertext link, pada saat diklik.
- Merancang form untuk transaksi, untuk pencarian informasi, formulir pemesanan, pemesanan produk, dan lain-lainnya.
- Melekatkan video klip, suara dan aplikasi lainnya dalam halaman web.

CSS adalah bahasa untuk mendeskripsikan tampilan dari halaman web, termasuk warna, layout dan tulisan. CSS dapat digunakan untuk tampilan dalam perangkat yang berbeda, misalnya layar yang lebar, layar kecil atau printer. CSS terpisah dari HTML dan dapat digunakan dengan bahasa XML [15].

2.2.3.4. Javascript & jQuery

Javascript adalah bahasa skrip yang ditempelkan pada kode HTML dan diproses di sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Sebagai contoh, dengan menggunakan JavaScript dimungkinkan untuk

memvalidasi masukan-masukan pada formulir sebelum formulir dikirimkan ke server. Javascript bukanlah bahasa Java dan merupakan dua bahasa yang berbeda. Javascript diinterpretasikan oleh klien (kodenya bisa dilihat pada sisi klien), sedangkan kode Java dikompilasi oleh pemrogram dan hasil kompilasinyalah yang dijalankan oleh klien. Struktur dari JavaScript adalah seperti Gambar 2.3 berikut:

```
<SCRIPT LANGUAGE = "JavaScript">
  <!--
    Penulisan kode javascript
  // -->
</SCRIPT>
```

Gambar 2. 3 Contoh kode Javascript

Kode `<!-- // -->` umumnya disertakan dengan tujuan agar sekiranya browser tidak mengenali JavaScript maka browser akan memperlakukannya sebagai komentar sehingga tidak ditampilkan pada jendela browser. Sedangkan jQuery merupakan suatu framework (library) javascript yang menekankan interaksi antara javascript dan HTML, atau bisa disebut juga sebagai cara baru dalam menuliskan kode javascript. jQuery akan mempercepat dan meringkaskan library javascript sehingga dengan menggunakan jQuery kita dapat mempercepat perpindahan dokumen HTML, penanganan event (event handling), pembuatan animasi didalam web sehingga web kita tampak seperti flash, dan juga interaksi AJAX untuk pengembangan web yang modern dan cepat. JQuery memiliki slogan “write less, do more” yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan code, tapi dengan hasil yang lebih banyak. jQuery mempunyai lisensi dibawah GNU General Public License dan MIT License artinya jQuery merupakan aplikasi yang Open Source dan bebas di publikasikan oleh siapapun. Ukurannya juga cukup kecil yaitu sebesar 56 KB (compressed) sehingga menggunakan jQuery

menjadi jauh lebih cepat dan mudah daripada menggunakan framework yang lain atau menggunakan Javascript konvensional. Beberapa hal yang bisa dilakukan dengan jQuery:

1. Mengakses bagian halaman tertentu dengan mudah.
2. Mengubah tampilan bagian halaman tertentu.
3. Mengubah isi dari halaman.
4. Merespon interaksi user dalam halaman dan menambahkan animasi ke halaman.
5. Mengambil informasi dari server tanpa me-refresh seluruh halaman.
6. Menyederhanakan penulisan Javascript biasa [6].

2.2.3.5. PHP

PHP merupakan bahasa pemrograman skrip yang diletakkan dalam server yang biasa digunakan untuk membuat aplikasi web yang bersifat dinamis. PHP mendukung berbagai database. Dengan demikian database yang dibuat dapat diakses oleh PHP dan memungkinkan untuk menampilkan isinya atau bahkan memanipulasi datanya melalui halaman web [15].

2.2.3.6. Web Server

Web Server adalah sebuah layanan internet yang telah disediakan oleh komputer. Dalam memasukkan web yang telah dirancang dalam internet, maka terlebih dahulu memiliki ruangan dalam internet dan ruangan inilah disediakan oleh server. Selain itu juga, Pengertian Web server merupakan sebuah perangkat komputer yang digunakan untuk mengakses segala jenis file yang terdapat pada halaman web melalui HTTP/HTTPS dengan menggunakan aplikasi atau program tertentu (dikenal dengan nama web browser) dan mengirimkan kembali hasil permintaannya dalam bentuk halaman – halaman web yang berbentuk dokumen HTML. Berikut contoh dari web server:

Apache merupakan Aplikasi Web Server yang paling banyak digunakan saat ini, selain dapat berjalan di banyak Operating System seperti Windows, Linux dan lainnya, Apache juga bersifat Open Source

Ada beberapa aplikasi Apache yang dapat berjalan di Windows dan di Linux yaitu:

- XAMPP
- WAMPP

Kelebihan Web Server Apache:

1. Apache termasuk dalam kategori freeware.
2. Apache mudah sekali proses instalasinya jika dibanding web server lainnya seperti NCSA, IIS, dan lain-lain.
3. Mampu beroperasi pada berbagai platform sistem operasi.
4. Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi.
5. Mudah dalam menambahkan peripheral lainnya ke dalam platform web servernya.

Kekurangan Web Server Apache:

1. Web server Apache tidak memiliki kemampuan mengatur load seperti IIS, sehingga akan terus mem-fork proses baru hingga nilai MaxClients tercapai atau hingga batas yang diizinkan oleh OS. Ini tentunya menguntungkan penyerang karena habisnya RAM akan lebih cepat tercapai.
2. Apache tidak memproses karakter kutip dalam string Referrer dan UserAgent yang dikirimkan oleh Client. Ini berarti Client dapat memformulasi inputnya secara hati hati untuk merusak format baris log akses.
3. Terganggunya proses upload data, yang bisa menyebabkan software salah dalam menerjemahkan ukuran data yang masuk.

Apache tomcat adalah server web berbasis open source yang mendukung untuk penggunaan JSP (Java Server Pages). Secara default, server tomcat ini belum memiliki package admin, sehingga tidak ada akses untuk admin. Yang ada hanyalah tomcat manager untuk memdeploy aplikasi web melalui file .war dari java web.

Ketika kita berusaha masuk ke URL admin ada keterangan Tomcat's administration web application is no longer installed by default. Download and install the "admin" package to use it. Hal itu dikarenakan tidak ada package admin dalam apache tomcat tersebut. Akses administration server tomcat ini bisa untuk setting JNDI. Java Naming and Directory Interface (JNDI) adalah sebuah API yang mendeskripsikan library Java yang standar untuk mengakses layanan naming dan directory seperti Domain Naming Service (DNS), dan Lightweight Directory Access Protocol (LDAP).

Apache Tomcat merupakan implementasi software open source dari Java Servlet dan JavaServer Pages teknologi. Java Servlet dan JavaServer Pages spesifikasi yang dikembangkan di bawah Java Community Process.

Kelebihan Apache Tomcat:

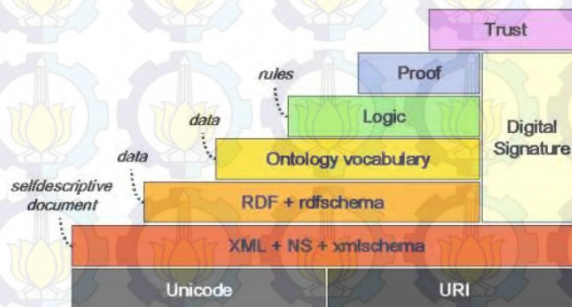
1. Performanya yang tinggi, stabil, memiliki banyak fitur
2. Mudah Dikonfigurasi
3. Menggunakan hanya sedikit sumber daya pada server.
4. Tidak bergantung kepada thread untuk melayani klien.

Kekurangan Apache Tomcat:

1. Belum support IPV6
 2. Update/patchnya versi terbarunya lama keluar
 3. Fast-CGInya tidak berfungsi maksimal
 4. Pemakainya tidak sebanyak Apache atau IIS
 5. (Komunitasnya tidak sebanyak Apache atau IIS)
- [15].

2.2.4. Semantic Web

Web Semantik (*semantic web*) adalah perkembangan generasi web berikutnya atau yang bisa disebut sebagai evolusi dari WWW (World Wide Web), yang dicetuskan pada tahun 2002. Web semantik didefinisikan sebagai sekumpulan teknologi, dimana memungkinkan komputer memahami arti dari sebuah informasi berdasarkan metadata, yaitu informasi mengenai isi informasi [8]. Dengan adanya metadata, komputer diharapkan mampu mengartikan hasil pemasukan informasi sehingga hasil pemasukan informasi sehingga hasil pencarian menjadi lebih detail dan tepat. W3C (World Wide Web Consortium) mendefinisikan format metadata tersebut adalah RDF (Resource Description Format). Tiap unit dari RDF adalah tiga komposisi, yaitu subject, predicate, dan object. Subject dan predicate adalah entitas yang ditunjukkan oleh teks. Sedangkan predicate adalah komposisi yang menerangkan sudut pandang dari subject yang dijelaskan object. Hal yang paling menarik dari RDF yaitu object dapat menjadi subject yang nantinya diterangkan oleh object yang lainnya. Sehingga object atau masukan dapat diterangkan secara jelas dan detail, serta sesuai dengan keinginan pengguna yang menggunakan masukan. Pada Gambar 2.4 berikut menjelaskan mengenai layer pada web semantic.



Gambar 2. 4 Layer Web Semantic

- Uniform Resource Identifier (URI) merupakan penamaan yang unik untuk identifikasi web semantik;
- Unicode mempresentasikan dan memanipulasi teks ke banyak bahasa;
- XML sangat cocok dalam pengiriman dokumen melalui web;
- RDF menyediakan sebuah pemaknaan sederhana untuk model data, yang dapat dipresentasikan dalam sintaks XML;
- RDF Schema menyediakan dasar-dasar kosa kata untuk RDF yang memungkinkan untuk membuat hirarki kelas dan propertinya;
- Ontologi vocabulary memperluas RDF Schema dengan menambahkan konsep yang lebih canggih untuk menambahkan sebuah batasan, seperti kardinalitas, batasan nilai, karakteristik dari properti seperti transitive. Ini didasarkan pada logika sehingga memberikan kekuatan reasoning pada web semantik;
- Logic digunakan untuk meningkatkan bahasa ontology lebih lanjut dan mengizinkan penulisan dari deklarasi pengetahuan;
- Proof melibatkan proses pengurangan nyata seperti halnya penyajian bukti di bahasa web dan validasi bukti;
- Trust bertujuan untuk memastikan dan memverifikasi bahwa pernyataan web semantik berasal dari sumber yang terpercaya. Ini dapat dicapai dengan tepat menggunakan 'digital signature' dari pernyataan RDF [10].

Dalam kesepakatan bersama mengenai *semantic web* mengatur beberapa kesepakatan sebagai berikut:

1. Linked Data
Data pada *semantic web* harus saling terhubung. Kesepakatan bersama mengenai format data adalah RDF, jadi data diubah ke bentuk RDF terlebih dahulu.

2. Vocabularies

Kesepakatan yang lain mengenai vocabularies dan ontology adalah menggunakan yang sudah ada jika tidak ada dipertimbangkan dengan vocab yang berhubungan sehingga terjadi kesamaan. Salah satu vocab yang ada adalah SKOS dan FOAF.

3. Query

Selanjutnya dalam masalah query disepakati menggunakan query SPARQL. Akan di bahas pada sub bab selanjutnya.

4. Inference

Dalam proses pengambilan keputusan dan kesimpulan pada data beserta aturan dalam data tersebut digunakan metode RIF dan OWL, yang memungkinkan menangani dalam proses *reasoning* [20].

Untuk membangun web semantik harus memenuhi persyaratan di bawah ini:

- Mendukung RDF, RDF-Schema dan juga mendukung OWL (Web Ontology Language).
- Dukungan untuk SPARQL dan juga dukungan spesifikasi SPARQL Extension seperti insert, hapus, tambah, dan update.
- Dapat menyimpan RDF skala data besar secara efisien.
- Memiliki kemampuan inferensi untuk ontologi OWL.
- Dapat menyebarkan Linked Data [1].

2.2.4.1. Resource Description Frame (RDF)

Resource Description Frame (RDF) merupakan bagian utama dalam *semantic web* yaitu format untuk menyediakan informasi dalam bentuk yang dapat dibaca oleh mesin. Menggunakan URL dan string teks sebagai istilah, yang merupakan sebuah model sederhana untuk mendeskripsikan hubungan antar

resource. Dalam memberi arti RDF dinyatakan melalui keterangan berupa schema. Schema ini yang dianggap sebagai kamus atau aturan baku. Schema ini mendefinisikan istilah-istilah yang digunakan dalam pernyataan RDF dan memberikan arti tertentu untuk istilah dalam sumber daya tersebut [10].

RDF menyediakan model data yang sangat sederhana di satu sisi tapi sangat disesuaikan terhadap arsitektur web di sisi lainnya. Dalam RDF, sebuah deskripsi dari sumber direpresentasikan sebagai sejumlah triple. Tiga bagian dari setiap triple disebut subject, predicate dan object. Sebuah struktur dasar triple dari kalimat sederhana, sebagai berikut:

Chris	has the email address	chris@bizer.de
(subject)	(predicate)	(object)

Subject dari triple adalah URI yang mendefinisikan sumber. Object dapat berupa nilai literal sederhana, seperti string, numerik, atau tanggal, atau URI dari sumber daya lainnya yang berkaitan dengan subject. Predicate mengindikasikan hubungan diantara subject dan predicate, misalnya ini adalah nama atau tanggal lahir. Predicate juga berupa URI. URI predicate didapat dari kamus data (vocabularies), yaitu sejumlah URI yang dapat digunakan untuk merepresentasikan informasi tentang domain tertentu. RDF adalah ide dasar dalam mendefinisikan sesuatu menggunakan URI, dan mendeskripsikan sumber daya dalam istilah-istilah dari properti sederhana dan nilai properti. Ini memungkinkan RDF untuk merepresentasikan pernyataan-pernyataan sederhana tentang sumber daya sebagai graph dari node dan arc yang merepresentasikan sumber daya, dan propertipropertinya serta nilainya. Pernyataan berikut “seseorang yang diidentifikasi dengan <http://www.w3.org/People/EM/contact#me>, yang bernama Eric Miller, memiliki alamat email em@w3.org dan memiliki gelar Dr”, dapat direpresentasikan dalam RDF graph pada Gambar 2.5 berikut:



Gambar 2. 5 RDF Graph Mendeskripsikan Eric Miller

RDF juga menyediakan sebuah sintaks berbasis XML (disebut juga RDF/XML) untuk menyimpan dan pertukaran data [15].

2.2.4.2. Ontologi

Ontologi adalah representasi simbolis tentang pengetahuan objek, kelas objek, properti objek, dan relasi antar objek untuk merepresentasikan suatu pengetahuan tentang domain aplikasi [2]. Ontology merupakan definisi dari pengertian dasar dan relasi vocabulary yang berasal sebuah area sebagaimana aturan dari kombinasi istilah dan relasi untuk mendefinisikan, vocabulary ontology ini memiliki potensi untuk digunakan dalam menjelaskan pengetahuan pada suatu domain. Ontology juga sangat penting karena dapat digunakan untuk menerangkan tentang struktur suatu disiplin ilmu. Ontology memberikan kemungkinan untuk berpindah dari pandangan berorientasi dokumen, kearah pengetahuan yang saling terkait, dikombinasikan dan untuk digunakan di waktu kemudian [7].

Ontologi memudahkan dalam berbagi pengetahuan dan penggunaan kembali yakni suatu pemahaman umum dari berbagai jenis konten yang menjembati aplikasi dan manusia. Secara teknis, ontologi adalah suatu potongan pengetahuan secara teks, yang diletakan di web agar agent dapat berkonsultasi ke ontologi ketika diperlukan, dan mempresentasikan menggunakan sintak dari suatu penyajian bahasa ontologi. Ontologi sendiri dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah. Deskripsi tersebut berisi kelas, properti, dan instance. Deskripsi ini dapat membantu sistem komputer dalam menggunakan istilah-istilah tersebut dengan cara yang lebih mudah [10].

2.2.4.3. Ontology Web Language (OWL)

Ontology Web Language (OWL) adalah suatu bahasa yang dapat digunakan oleh aplikasi-aplikasi yang bukan sekedar menampilkan informasi pada pengguna melainkan juga perlu memproses isi informasi itu. OWL dapat digunakan untuk merepresentasikan arti serta makna dari istilah-istilah suatu domain pengetahuan secara eksplisit dan relasi antar istilah tersebut. OWL juga memiliki fasilitas yang lebih dibanding XML, RDF, RDFS dalam mengekspresikan arti secara semantic. OWL juga melebihi bahasa-bahasa lain tersebut diatas dalam kemampuan merepresentasikan isi dokumen yang dapat diinterpretasi oleh mesin [7].

Awal tahun 2004 W3C resmi melepaskan OWL sebagai bahasa rekomendasi untuk mempresentasikan ontologi. OWL dikembangkan dimulai dari uraian logika dan DAM+OIL. OWL adalah satu set dari elemen dan atribut XML, mendeskripsikan arti dan relasi diantaranya [14]. OWL adalah suatu bahasa yang dapat digunakan oleh aplikasi-aplikasi yang bukan sekedar menampilkan informasi tersebut pada manusia, melainkan juga yang perlu memproses isi informasi isi. Dengan menggunakan OWL, dapat menambah

kosakata tambahan disamping semantik formal yang telah dibuat sebelumnya menggunakan XML, RDF, dan RDF Schema. Hal ini sangat membantu penginterpretasian mesin yang lebih baik terhadap isi web [10].

2.2.4.4. SPARQL

SPARQL adalah bahasa query yang digunakan untuk mengakses data RDF dan RDFS. Protokol SPARQL untuk spesifikasi RDF adalah mendefinisikan protokol remote untuk melakukan query dan menerima hasil. Hasil dari query SPARQL dapat berupa format XML yang mendefinisikan representasi dari hasil query SPARQL. SPARQL didasarkan pada pencocokan pola graph terhadap graph RDF. Pola graph dasar adalah satu set pola triple ditulis dalam urutan pola triple. Data akan direpresentasikan dalam format Turtle. Klausa “SELECT” dan “WHERE” adalah sama seperti dalam SQL, tapi SPARQL dan SQL adalah dua bahasa yang berbeda.

Variabel dapat ditulis dengan ?x [15]. Hasil dari query SPARQL dapat mengembalikan nilai dalam beberapa format data, antara lain : XML, JSON, RDF, dan HTML. Untuk menjalankan SPARQL, dapat menggunakan beberapa tools dan APIs seperti: ARQ, Rasqal, RDF::Query, twingql, Pellet, dan KAON2. Tools tersebut memiliki API yang memungkinkan pemrogram untuk memanipulasi hasil query dengan berbagai aplikasi yang ada. Namun, sebagai standar dapat menggunakan SPARQL Query Results XML Format yang direkomendasikan oleh W3C [10].

Secara umum RDF graph dapat dijelaskan dengan tabel beserta kolomnya, jadi setiap baris merepresentasikan subjek, predikat, dan objek. Query SPARQL juga dapat digambarkan ke dalam aljabar relasional. Misalkan untuk query seperti Gambar 2.6 di bawah ini.

```

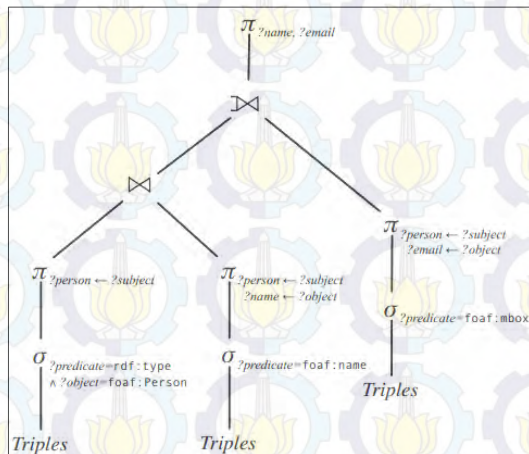
SELECT ?name ?email
WHERE {
  ?person rdf:type foaf:Person .
  ?person foaf:name ?name .
  OPTIONAL { ?person foaf:mbox ?email }
}

```

Gambar 2. 6 Contoh query SPARQL

Query di atas dapat dijelaskan sebagai mengambil nama dan email dari orang, dimana nama merupakan nilai dari foaf:name dari Person serta email dari foaf:mbox Person, dan email opsional maksudnya jika ada nilai mbox akan diambil, namun jika tidak maka nilai mbox tidak diambil.

Lebih jelasnya dapat dilihat pada Gambar 2.7 di bawah ini.



Gambar 2. 7 Representasi query SPARQL ke dalam aljabar relasional

Aljabar relasional adalah sebuah bahasa query prosedural yang terdiri dari sekumpulan operasi dimana masukannya adalah satu atau dua relasi dan keluarannya adalah sebuah relasi baru sebagai hasil dari operasi tersebut. Aljabar relasional

merupakan representasi dari setiap query agar orang lain mengetahui maksud dari suatu query. Seperti halnya dalam query bahasa lain, dalam SPARQL juga dapat digambarkan ke dalam aljabar relasional. Seperti pada Gambar 2.7 di atas.

Tupel merupakan kumpulan elemen data, atau kumpulan beberapa nilai atribut, atau kumpulan field-data yang mewakili satu entitas secara lengkap. RDF Tuple merupakan bagian dari RDF Term. RDF Tuple menyebutkan data pada entitas tertentu, misalnya.

$$t = \begin{cases} ?person & \rightarrow \langle \text{http://example.org/people\#Bob} \rangle \\ ?name & \rightarrow \text{"Bob"} \\ ?email & \rightarrow \langle \text{mailto:bob@example.org} \rangle \\ ?age & \rightarrow \text{"42"} \end{cases}$$

Gambar 2. 8 Contoh RDF Tuple

Sedangkan pada RDF Term harus terdiri dari triple yaitu terdiri ?subject ?predicate ?object, seperti pada Gambar 2.9 di bawah ini.

$$t = \begin{cases} ?subject & \rightarrow \langle \text{http://example.org/people\#Bob} \rangle \\ ?predicate & \rightarrow \text{ex:name} \\ ?object & \rightarrow \text{"Bob"} \end{cases}$$

Gambar 2. 9 Contoh RDF Term/Triple

Selanjutnya adalah Relasi RDF yang merupakan kumpulan dari banyak RDF Tuple, digambarkan dalam bentuk tabel dengan setiap kolom sebagai property atau variable. Contohnya pada Gambar 2.10 di bawah ini.

<i>?person</i>	<i>?name</i>	<i>?email</i>	<i>?age</i>
ex:Alice	"Alice"		"33"
ex:Bob	"Bob"	<mailto:bob@example.org>	"42"

Gambar 2. 10 Contoh Relasi RDF

Sehingga jika digambarkan ke dalam bentuk RDF graph adalah seperti pada Gambar 2.11 di bawah ini.

<i>?subject</i>	<i>?predicate</i>	<i>?object</i>
ex:Alice	ex:name	"Alice"
ex:Alice	ex:age	"33"
ex:Bob	ex:name	"Bob"
ex:Bob	ex:mbox	<mailto:bob@example.org>
ex:Bob	ex:age	"42"

Gambar 2. 11 RDF Graph

Berikut aljabar relasional yang umum digunakan serta yang ditranslasikan pada query SPARQL.

1. Selection

Operasi selection dapat didefinisikan sebagai "Kumpulan semua tuple-tuple/record-record dalam suatu tabel yang memenuhi kondisi P". Operasi selection berfungsi untuk menyeleksi tuple-tuple yang memenuhi predikat yang diberikan dari sebuah tabel relasi. Simbol sigma " σ " digunakan untuk menunjukkan operasi select. Predikat muncul sebagai subscript dari σ dan kondisi yang diinginkan yang ditulis dalam predikat.

Contoh penggunaannya, adalah seperti Gambar 2.12 di bawah ini.

$$\sigma_{?predicate=ex:name}(R)$$

Gambar 2. 12 Selection

Maksud dari aljabar relasional di atas adalah tampilkan daftar R yang memiliki ?memiliki ?predicate ex:name.

Contoh pada query SPARQL adalah seperti pada Gambar 2.13 di bawah ini.

```
SELECT *
WHERE {
    ?subject ex:name ?object.
}
```

Gambar 2. 13 Contoh selection pada SPARQL

2. Projection

Operasi projection berfungsi untuk memilih nilai atribut-atribut tertentu saja dari sebuah tabel relasi. Simbol phi “ π ” digunakan untuk menunjukkan operasi projection. Predikat muncul sebagai subscript dari π dan hanya nama atribut yang diinginkan yang ditulis dalam predikat.

Contohnya adalah seperti pada Gambar 2.14 di bawah ini.

$$\pi_{?person \leftarrow ?subject \atop ?name \leftarrow ?object}(R)$$

Gambar 2. 14 Projection

Maksud dari relasi aljabar di atas adalah mengambil ?person dan ?name yang mana ?person merupakan subject dari R dan ?name ?object dari R. Contoh pada SPARQL adalah seperti pada Gambar 2.15 di bawah ini.

```
SELECT ?person ?name
WHERE {
    ?person ?predicate ?name.
}
```

Gambar 2. 15 Contoh projection pada SPARQL

3. Join

Operasi inner join/natural join berfungsi untuk menggabungkan operasi selection dan cartesian product menjadi hanya 1 operasi saja. Simbol “ $\bowtie$ ” digunakan untuk menunjukkan operasi natural join. Operasi natural join hanya menghasilkan tuple yang mempunyai nilai yang sama pada 2 atribut yang bernama sama pada 2 tabel relasi yang berbeda. Sedangkan LEFT outer join “ $\leftarrow\bowtie$ ” menampilkan semua variable pada tuple pertama yang dipilih. Misalkan ada dua tuple A dan B.

A		B	
?person	?name	?person	?parent
ex:Alice	"Alice"	ex:Bob	ex:Charles
ex:Bob	"Bob"	ex:Bob	ex:Dorothy

Gambar 2. 16 Contoh tuple A dan B

$A \bowtie B$, menjadi:

$A \bowtie B$		
<i>?person</i>	<i>?name</i>	<i>?parent</i>
ex:Bob	"Bob"	ex:Charles
ex:Bob	"Bob"	ex:Dorothy

Gambar 2. 17 Inner join

$A \bowtie B$, menjadi:

$A \bowtie B$		
<i>?person</i>	<i>?name</i>	<i>?parent</i>
ex:Alice	"Alice"	
ex:Bob	"Bob"	ex:Charles
ex:Bob	"Bob"	ex:Dorothy

Gambar 2. 18 Left outer join

Maksudnya adalah menampilkan *?person* dan *?name* pada tuple A dan *?person* dan *?parent* pada tuple B. yang membedakan natural join dan left outer join adalah pada kondisi kedua tuple, jika pada natural join data yang ditampilkan yang memiliki kesamaan pada kedua tuple, sedangkan left outer join untuk tuple pertama yang dipilih ditampilkan semua. Inner join pada SPARQL dapat menggunakan FILTER maupun tidak. Pada left outer join menggunakan OPTIONAL, yang berarti menampilkan semua variable pada tuple awal.

Sehingga untuk query SPARQL menjadi:

$A \bowtie B$:

```

SELECT ?person ?name ?parent
WHERE {
    ?person ex:name ?name.
    ?person ex:parent ?parent.
}

```

Gambar 2. 19 Contoh Inner join pada SPARQL

A $\bowtie$ B :

```

SELECT ?person ?name ?parent
WHERE {
    ?person ex:name ?name.
    OPTIONAL
    {
        ?person ex:parent ?parent.
    }
}

```

Gambar 2. 20 Contoh left outer join pada SPARQL

4. Union

Operasi union berfungsi untuk mendapatkan gabungan nilai atribut dari sebuah tabel relasi dengan nilai atribut dari tabel relasi lainnya. Simbol “U” digunakan untuk menunjukkan operasi union. Misalkan ada dua tuple A dan B.

A		B	
<i>?person</i>	<i>?name</i>	<i>?person</i>	<i>?parent</i>
ex:Alice	"Alice"	ex:Bob	ex:Charles
ex:Bob	"Bob"	ex:Bob	ex:Dorothy

Gambar 2. 21 Tuple A dan B

A U B, menjadi:

$A \cup B$		
<i>?person</i>	<i>?name</i>	<i>?parent</i>
ex:Alice	"Alice"	
ex:Bob	"Bob"	
ex:Bob		ex:Charles
ex:Bob		ex:Dorothy

Gambar 2. 22 A Union B

Dalam query SPARQL adalah:

```
SELECT ?person ?name ?parent
WHERE {
  {
    ?person ex:name ?name.
  }
  UNION
  {
    ?person ex:parent ?parent.
  }
}
```

Gambar 2. 23 Contoh union pada SPARQL

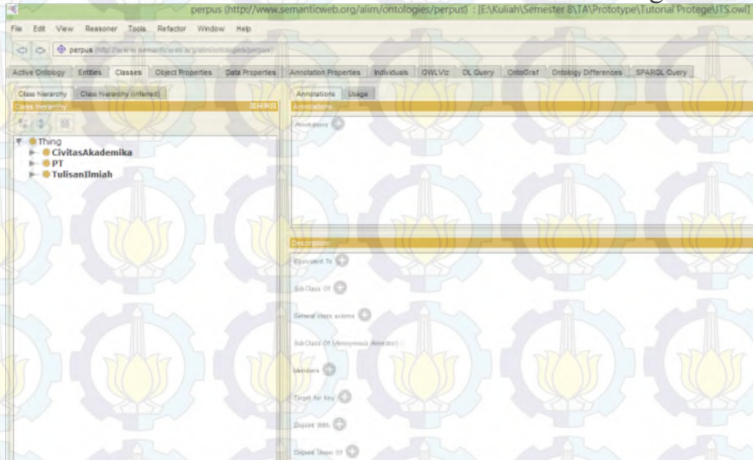
Berarti menampilkan ?person dan ?name pada tuple A UNION pada ?person dan ?parent pada tuple B [21].

2.2.4.5. SPARQL PHP-Lib

SPARQL PHP-Lib merupakan library untuk bahasa pemrograman PHP dalam menangani query SPARQL. Fungsi utama dari SPARQL PHP-Lib ini adalah untuk melakukan parsing data RDF dengan berbagai format: N-Triples, RDF/JSON, RDF/XML, Turtle. SPARQL PHP-Lib juga mendukung penanganan query SPARQL [16].

2.2.4.6. Protégé

Protégé adalah editor ontologi open source dan framework berbasis pengetahuan. Platform Protégé mendukung dua langkah utama dalam pemodelan ontologi, yaitu editor Protégé-Frames dan editor Protégé-OWL.



Gambar 2. 24 Contoh Antarmuka Protege

Ontologi Protégé dapat di-exported kedalam berbagai format termasuk RDFS, OWL dan XML Schema. Protégé adalah berbasis Java, extensible, dan menyediakan lingkungan plug-and-play yang membuatnya menjadi basis yang fleksibel untuk prototyping cepat dan pengembangan aplikasi. Beberapa yang dapat dilakukan Protégé adalah:

- Load dan save OWL dan RDF ontologi
- Edit dan visualisasi kelas-kelas, propertiproperti dan aturan SWRL.
- Mendefinisikan karakterisasi kelas logikal sebagai ekspresi OWL.
- Mengeksekusi sebab akibat seperti description logic classifier

- Edit individual OWL untuk semantik web [11].

2.2.4.7. OpenRDF Sesame

Sesame adalah sebuah kerangka kerja dalam melakukan manipulasi serta pemrosesan data berbentuk RDF. OpenRDF Sesame mendukung untuk melakukan penyimpanan, pengolahan serta transfer data RDF. OpenRDF Sesame dibangun dengan pemrograman Java, sehingga kita perlu untuk menginstal package Sesame ke web server yang mendukung JSP, salah satunya adalah Apache Tomcat.

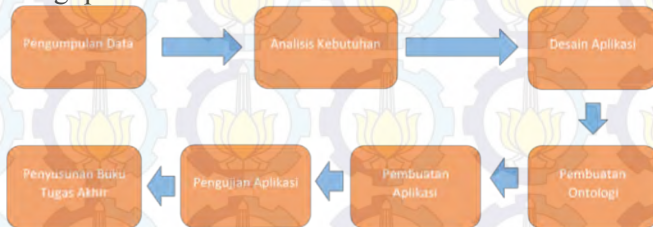
2.2.5. DBpedia

DBpedia merupakan sebuah proyek yang memiliki tujuan mengambil konten terstruktur dari informasi yang tercipta di Wikipedia. Informasi terstruktur ini tersedia di World Wide Web. DBpedia memungkinkan pengguna mencari hubungan dan properti yang berkaitan dengan sumber daya Wikipedia, termasuk tautan ke dataset lainnya. DBpedia disebut Tim Berners-Lee merupakan salah satu bagian proyek data tertaut atau linked-data paling terkenal. Latar belakang Proyek ini dirintis oleh beberapa orang di Free University of Berlin dan University of Leipzig, bekerja sama dengan OpenLink Software. Dataset publik pertamanya diterbitkan tahun 2007. Proyek ini berada di bawah lisensi gratis dan mengizinkan pengguna lain memakai ulang datasetnya. Sebagian besar isi artikel Wikipedia adalah teks bebas, tetapi ada juga informasi terstruktur yang tertanam di dalamnya, seperti tabel "kotak info" (panel di kanan atas artikel Wikipedia), informasi kategorisasi, gambar, koordinat geografis dan pranala luar ke halaman web luar. Informasi terstruktur ini diambil dan diletakkan dalam dataset seragam yang dapat dicari pengguna. Penggunaan DBpedia memiliki cakupan pengetahuan manusia yang luas. Keunggulan ini menjadikannya penghubung alami untuk menghubungkan dataset, tempat dataset luar dapat tertaut pada konsep-konsepnya.

BAB III

METODOLOGI PENELITIAN

Bagian ini menjelaskan metodologi yang digunakan dalam pengerjaan tugas akhir ini. Metodologi ini diperlukan sebagai panduan secara terstruktur dan sistematis dalam pengerjaan tugas akhir. Gambaran dari metodologi bisa dilihat pada Gambar 3.1. Berikut penjelasan dari beberapa tahap yang ada dalam metodologi penelitian.



Gambar 3. 1 Metodologi Penelitian

3.1. Pengumpulan Data

Langkah ini dilakukan agar penulis dapat lebih mengerti akan topik yang akan dibahas. Adapun beberapa tahapan dalam langkah ini yaitu:

- 1) Konseptualisasi permasalahan
Penulis meneliti dan merumuskan masalah serta menentukan kebutuhan pengguna dari sistem yang dibangun.
- 2) Studi pustaka
Tahapan ini dilakukan dengan mengumpulkan referensi dan sumber bacaan yang berhubungan dengan sistem yang akan dibangun, melakukan pembelajaran serta pemahaman mengenai permasalahan yang ada, seperti mengenai tanaman obat yang ada di Indonesia, jenis-jenis penyakit yang dapat diobati dengan tanaman obat, teknologi *semantic web*: owl, rdf serta pengembangan aplikasi ensiklopedia untuk tanaman obat.

3) Pengumpulan data

Pengumpulan data-data ini dapat diambil dari jurnal ilmiah, artikel, buku referensi, aplikasi lain, dan halaman web. Pengumpulan data dilakukan terhadap tanaman obat. Dari studi pustaka dijelaskan bahwa dari sekian banyak jenis tanaman obat yang ada baru sekitar 300 jenis tanaman obat yang dimanfaatkan. Sehingga pengumpulan data ini adalah mengumpulkan informasi tanaman obat sebanyak 300 jenis.

3.2. Analisis Kebutuhan

Pada tahap ini dilakukan analisis terhadap beberapa hal terkait kebutuhan sistem, fungsional, pengguna, dan cara kerja aplikasi. Diharapkan aplikasi yang akan dibangun sesuai dengan kebutuhan untuk memecahkan permasalahan. Kegiatan yang dilakukan pada tahapan ini adalah sebagai berikut:

1. Analisis masalah

Melakukan analisis terhadap masalah-masalah yang dihadapi pada sistem berjalan serta menentukan tujuan pengembangan sistem dari hasil analisis masalah tersebut.

2. Analisis kebutuhan aplikasi

Melakukan analisis terhadap kebutuhankebutuhan dari aplikasi yang akan dibangun, yaitu berhubungan dengan kelakuan dari aplikasi serta fitur-fitur yang disediakan aplikasi.

3. Analisis kebutuhan konten

Menentukan sumber-sumber data yang akan digunakan dalam aplikasi ini, data yang digunakan adalah data yang sebelumnya telah tersedia di internet.

Setelah melakukan analisis permasalahan dan kebutuhan diperlukan suatu pedoman untuk membangun aplikasi ini agar aplikasi ini dapat dikatakan telah menerapkan teknologi *semantic web* dan berbentuk ensiklopedia. Untuk dapat dikatakan sudah

menggunakan teknologi *semantic web* adalah harus sudah memenuhi syarat sebagai berikut:

1. Mendukung RDF, RDF-Schema dan juga mendukung OWL (Web Ontology Language).
2. Dukungan untuk SPARQL dan juga dukungan spesifikasi SPARQL Extension seperti insert, hapus, tambah, dan update.
3. Dapat menyimpan RDF skala data besar secara efisien.
4. Memiliki kemampuan inferensi untuk ontologi OWL.
5. Dapat menyebarkan Linked Data

Sedangkan untuk dapat dikatakan sudah berbentuk ensiklopedia adalah harus sudah memenuhi syarat sebagai berikut:

1. Adanya artikel/topik, sub topik.
2. Adanya definisi artikel/topik dan diikuti penjelasan umum.
3. Adanya rujuk silang (cross reference) atau futher more, see also.
4. Adanya paragraf, ilustrasi, gambar, grafik, table time line.
5. Tersusun dengan baik, contohnya disajikan secara sistematis alfabetis (A – Z), atau tematis, historis, kronologis.
6. Adanya indeks.

3.3. Desain Aplikasi

Pada tahap ini kebutuhan yang telah didapat dari tahapan sebelumnya berkembang menjadi fitur dan fungsi system untuk dikembangkan lagi menjadi rancangan serta desain dari aplikasi. Tahapannya adalah perancangan arsitektur sistem aplikasi ensiklopedia. Pada tahapan ini penulis memodelkan desain dari arsitektur sistem secara keseluruhan. Arsitektur sistem bertujuan untuk merepresentasikan cara kerja sistem dan pertukaran data secara menyeluruh. Aplikasi ini dikembangkan dengan metode

pengembangan ICONIX dengan pemodelan menggunakan konsep UML. Keluaran dari metode pengembangan ICONIX berupa Domain Model, GUI Story Board, Use Case Model, Robustness Diagram, Sequence Diagram, dan Class Diagram.

3.4. Pembuatan Ontologi

Pada tahapan ini penulis melakukan pemodelan terhadap ontologi aplikasi, yaitu mendefinisikan taksonomi untuk aplikasi web semantik yang akan dibangun. Pada tahap perancangan ini dipilih bahasa ontologi yaitu OWL, untuk alat bantu dalam merancang ontologi digunakan Protégé. Keluarannya adalah desain ontologi tanaman obat. Berikut langkah-langkahnya dalam membangun ontologi dengan berpedoman pada Methontology [9].

1. Specification
Melakukan identifikasi terhadap tujuan-tujuan dari ontologi, serta kebutuhan-kebutuhan yang harus dipenuhi. Seperti pemilihan bahasa untuk ontologi.
2. Knowledge acquisition
Melakukan pengumpulan pengetahuan mengenai domain, dalam hal ini adalah melakukan studi literatur terhadap tanaman obat.
3. Conceptualisation
Mendefinisikan kelas ontologi dan menyusun kelas tersebut dalam hirarki taksonomi dengan menggunakan proses pengembangan dari tingkat teratas ke tingkat terendah, mendefinisikan properti, mendefinisikan facets pada slot atau axiom pada properti.
4. Integration
Mempertimbangkan ontologi yang sudah ada atau berhubungan dengan domain tanaman obat, dengan menggunakan kembali vocabulary yang sudah ada.
5. Implementation
Merancang konsep ontologi dan merepresentasikannya ke dalam bahasa ontologi yang dipilih, dalam hal ini akan

membuat instance jenis-jenis tanaman obat dan penyakit serta mengisi nilai slot pada instance.

6. Evaluation

Melakukan verifikasi serta validasi dari ontologi dengan memeriksa konsistensi logika, sintaks, dan semantik dari elemen-elemen yang ada pada ontologi. Pemeriksaan konsistensi dapat mencakup klasifikasi otomatis yang mendefinisikan konsep-konsep baru berdasarkan properti dan relasi [17].

7. Documentation

Melakukan pemeriksaan kembali dari setiap langkah yang dilakukan dan memastikan langkah yang dilakukan telah selesai.

3.5. Pembuatan Aplikasi

Pada tahap ini ada beberapa tahapan, sebagai berikut:

1. Metode pengumpulan konten dari sumber

Setelah dilakukan analisis, penulis menentukan untuk pengumpulan konten dari sumber dilakukan setiap kali pengguna menjalankan aplikasi, dengan estimasi ukuran data yang diambil tidak besar sehingga tidak membebani bandwidth.

2. Pembangunan endpoint

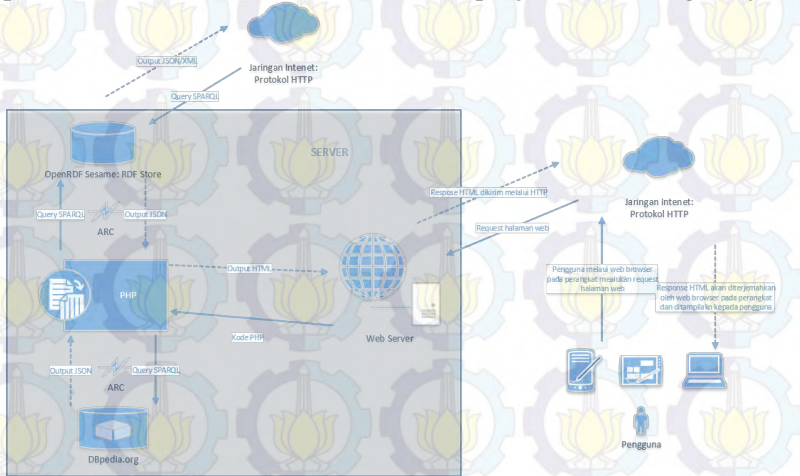
Pada aplikasi ini juga disediakan endpoint untuk ontologi tanaman obat. Sehingga nantinya ontologi tanaman obat ini dapat dimanfaatkan oleh orang lain, dan juga menjadi dari penelitian ini untuk berpartisipasi dalam visi web semantik yaitu menyediakan linked data ontologi tanaman obat agar dapat dimanfaatkan kembali oleh orang lain. Untuk pembangunan endpoint dapat memanfaatkan OpenRDF Sesame sebagai tool agar dapat diakses dari luar jaringan dengan SPARQL query.

3. Penulisan kode program

Penulisan kode perangkat lunak berdasarkan hasil dari desain sebelumnya dengan menggunakan bahasa pemrograman PHP,

Ajax, jQuery serta HTML. Kode utama menggunakan PHP untuk melakukan fungsi SPARQL query dengan menambahkan plugin SPARQL PHP-Lib, plugin ini berfungsi untuk melakukan query ke SPARQL endpoint dan memarsing hasil outputnya.

Lebih jelasnya berikut gambaran secara umum cara kerja aplikasi pada Gambar 3.2, beserta penjelasan singkatnya:



Gambar 3. 2 Gambaran Umum Cara Kerja Aplikasi

1. Pengguna mengakses halaman web ensiklopedia tanaman obat melalui web browser pada perangkatnya.
2. Web browser tersebut akan melakukan request HTTP melalui koneksi internet ke server dimana web ensiklopedia tersebut berada.
3. Web server yang terdapat pada server akan melakukan pengecekan terhadap perintah dari request tersebut yang kemudian akan menjalankan script PHP.
4. Untuk penanganan query SPARQL, maka plugin SPARQL PHP-Lib akan menghubungkannya ke OpenRDF Sesame yang menjadi penyimpanan dari RDF.

Jika query yang dilakukan tidak error maka OpenRDF Sesame mengirimkan data dengan format JSON sebagai output dari query.

5. Untuk informasi detail tanaman obat dan penyakit web server akan melakukan query ke DBpedia, dan DBpedia akan mengirimkan data yang diminta berbentuk JSON.
6. SPARQL PHP-Lib PHP akan melakukan parsing dari data JSON dan menambahkannya ke dalam kode HTML. Kemudian akan dikirimkan ke Web server.
7. Web server akan mengirimkan response ke client dalam bentuk HTML melalui protokol HTTP internet yang selanjutnya akan dikirimkan ke client.
8. Data yang berbentuk HTML tersebut yang telah sampai pada client akan diterjemahkan oleh web browser dan ditampilkan ke pada pengguna berbentuk halaman web yang sesuai dengan permintaan pengguna.
9. Server juga berguna sebagai endpoint, agar ontologi tanaman obat dapat diakses serta dimanfaatkan oleh banyak orang. Pengguna dapat melakukan query SPARQL untuk mendapatkan data. OpenRDF Sesame mengatur penanganan endpoint ontologi tersebut.

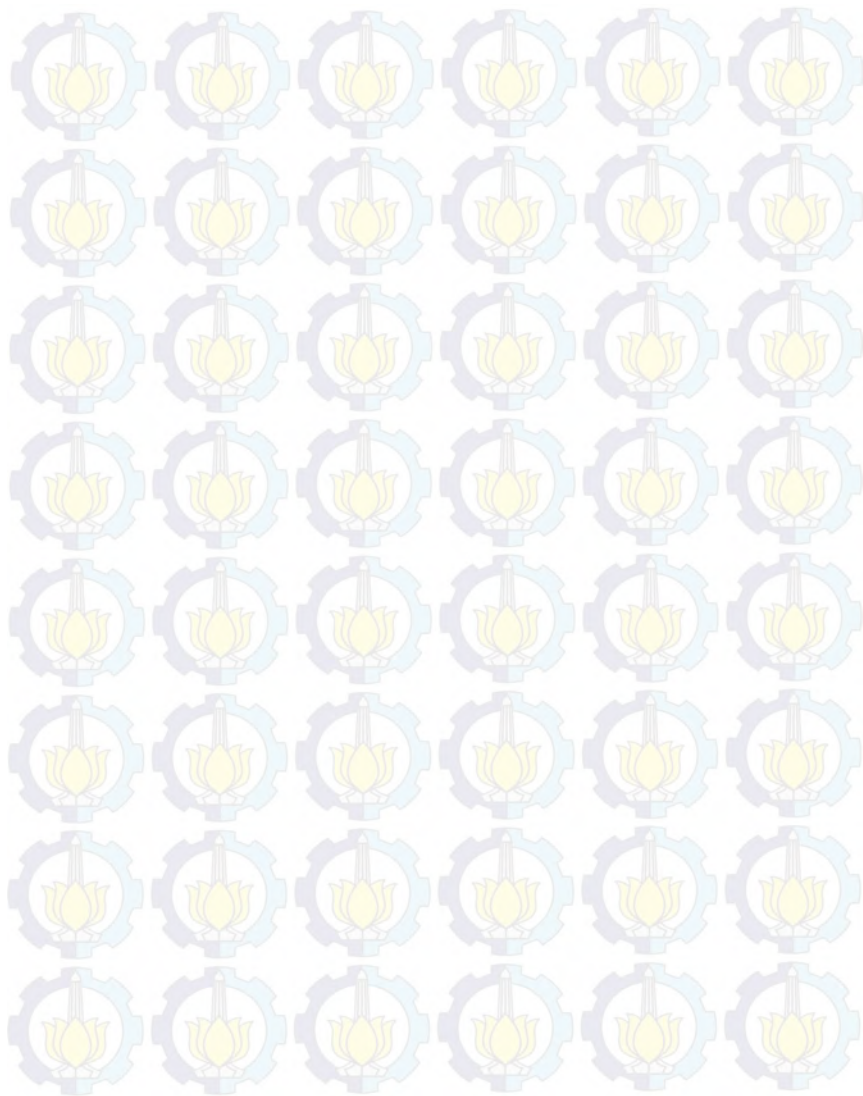
3.6. Pengujian Aplikasi

Pada tahap pengujian ini sistem diuji coba. Untuk pengujian sistem dilakukan dengan system testing mengetahui apakah kebutuhan sistem sudah terpenuhi, Melakukan pengecekan terhadap endpoint yang sudah dibuat untuk menguji apakah data sudah dapat diakses dari luar jaringan, serta melakukan *stress testing*.

3.7. Penyusunan Buku Tugas Akhir

Tahap terakhir dari penelitian ini adalah menyusun buku tugas akhir yang akan digunakan sebagai bahan evaluasi penelitian. Penyusunan buku tugas akhir ini disesuaikan dengan standar/template yang berlaku di Institusi.

(halaman ini sengaja dikosongkan)



BAB IV

ANALISIS KEBUTUHAN DAN DESAIN SISTEM

Pada bab ini dijelaskan proses analisis dan pembuatan desain aplikasi ensiklopedia tanaman obat. Desain sistem divisualisasikan dengan konsep Unified Modelling Language (UML), dan menggunakan model pengembangan ICONIX. Adapun yang dibahas pada bagian ini adalah spesifikasi kebutuhan, proses interaksi, arsitektur aplikasi, domain model, use case model, GUI prototype, robustness diagram, sequence diagram, class diagram, desain database, dan test case.

4.1. Deskripsi Umum Sistem

Secara umum dapat dijelaskan bahwa sistem aplikasi ini adalah aplikasi ensiklopedia tanaman obat berbasis web yang menerapkan teknologi web semantik yaitu ontologi. Penggunaan ontologi dimaksudkan agar data yang tersimpan dapat dikenali oleh mesin sehingga tercipta interaksi antara data yang saling terkait sehingga pengguna dapat melihat berbagai data yang disajikan dengan mudah.

Ontologi menjadi konsep dasar, dirancang sedemikian rupa sehingga domain dari tanaman obat dapat diimplementasikan dengan baik dalam sistem. Ontologi mendefinisikan *vocabulary* yang bersifat umum dan ditujukan oleh peneliti untuk membagikan informasi dalam domain tertentu. Ada beberapa alasan menggunakan ontologi yaitu: agar pengetahuan tentang domain dapat digunakan kembali, untuk membagikan pengetahuan dari struktur informasi tertentu, serta untuk menganalisa pengetahuan domain. Dalam perkembangan dunia web, W3C merancang suatu konsep RDF untuk merepresentasikan ontologi dalam halaman web, sehingga memungkinkan komputer dapat memahami informasi tersebut. Selanjutnya konsep web semantik sendiri, memungkinkan data yang telah dipublikasikan dapat digunakan kembali serta membuat data saling terhubung, W3C memberikan standar untuk

menggunakan query SPARQL. SPARQL memungkinkan dalam melakukan query dari banyak sumber yang berbeda, ini merupakan salah satu yang belum dapat dilakukan oleh bahasa query lain, Dalam studi kasus tugas akhir ini diperlukan penggunaan *semantic web* adalah karena dalam tugas akhir ini menggunakan sumber lain yaitu DBpedia digabungkan dengan ontologi yang dibuat, serta ontologi yang telah dibuat dapat di-*query* oleh orang lain. Konsep ensiklopedia sendiri merupakan konsep dalam menyajikan data agar pengguna dapat memahami dengan mudah. Ruang lingkup tanaman obat dalam aplikasi ensiklopedia ini akan menampilkan penjelasan mengenai tanaman obat tersebut meliputi nama ilmiah dan kandungan yang ada di dalamnya serta hubungan tanaman obat tersebut dengan penyakit dengan maksud tanaman obat tersebut dapat digunakan sebagai obat untuk penyakit tersebut. Diharapkan pengguna lebih mudah dalam mencari serta memahami tanaman obat dengan adanya aplikasi ensiklopedia ini. Serta juga dari ontologi yang sudah dibuat nantinya dapat di-*query* oleh pengguna lain yang membutuhkan data dari tanaman obat tersebut.

4.2. Analisis Kebutuhan

Kebutuhan terhadap pengembangan sistem aplikasi ensiklopedia tanaman obat ini akan dijelaskan pada bagian berikut dengan mengacu pada referensi dari buku tugas akhir “Pembuatan Aplikasi Ensiklopedia Makanan Tradisional Indonesia Berbasis Web Semantik” oleh Masrizal Frisnanto. Tidak lupa disesuaikan juga dengan aplikasi yang penulis buat. Analisis kebutuhan yang dilakukan meliputi kebutuhan pengguna dan kebutuhan fungsi.

4.2.1. Analisis Kebutuhan Pengguna Aplikasi

Pada referensi dijelaskan bahwa pengguna ada 3, yaitu administrator, kontributor dan pengguna umum, dikarenakan dalam aplikasi untuk ontologi makanan akan diubah dan ditambah oleh kontributor dan disetujui oleh administrator. Pada aplikasi yang akan penulis bangun ontologi tanaman obat tidak dapat diubah sehingga administrator dan kontributor tidak diperlukan. Analisis terhadap pengguna disimpulkan bahwa pengguna dalam aplikasi ini adalah pengguna umum atau dengan kata lain pengunjung saja, jadi siapa pun dapat mengakses aplikasi ensiklopedia tanaman obat ini. Adapun hal-hal yang dapat dilakukan oleh pengguna adalah sebagai berikut:

- Melihat informasi tanaman obat.
- Melihat informasi penyakit.
- Melakukan pencarian tanaman obat.

4.2.2. Analisis Kebutuhan Pengguna Ontologi

Analisis terhadap pengguna ontologi adalah pengguna yang ingin mengambil data tanaman obat dan diperlukan dalam mengembangkan aplikasi lain atau pun diintegrasikan dengan berbagai sumber ontologi. Adapun hal-hal yang dapat dilakukan oleh pengguna ontologi adalah melakukan query SPARQL 'SELECT' saja dimaksudkan untuk keamanan data dari perubahan data yang mungkin terjadi jika pengguna diizinkan untuk mengedit data. Untuk membantu pengguna dalam memahami ontologi tanaman obat dibuatkan front-end, sehingga lebih mudah dalam melakukan query SPARQL.

4.2.3. Fitur Aplikasi

Fitur aplikasi ini dan juga untuk penentuan proses pengembangan aplikasi ini hanya difokuskan pada pembuatan aplikasi ensiklopedia tanaman obat, untuk hal-hal lain yang diluar aplikasi tersebut hanya sebagai tambahan saja. Berdasarkan studi literatur dan pengumpulan data yang dilakukan secara keseluruhan

pengguna aplikasi dapat melihat dan mencari data. Yang membedakan adalah dalam aplikasi tersebut dimungkinkan kontributor dapat menambahkan data, namun karena pada aplikasi yang penulis buat tidak dimungkinkan perubahan serta penambahan data ontologi, maka fitur pada aplikasi ini hanya sebatas melihat dan mencari. Pada akhirnya dibuat suatu daftar kebutuhan dari pengguna yang disebutkan pada bagian sebelumnya. Pada Tabel 4.1 di bawah ini merupakan daftar fitur yang harus ada pada aplikasi ensiklopedia tanaman obat.

Tabel 4. 1 Daftar Fitur Aplikasi

KODE	Fitur
FA-01	Aplikasi dapat menampilkan daftar tanaman obat yang merupakan sub topik dari ensiklopedia tanaman obat.
FA-02	Aplikasi dapat menampilkan penjelasan umum mengenai tanaman obat, namun jika tidak ada informasinya maka menampilkan pesan tidak tersedianya informasi umum mengenai tanaman obat tersebut
FA-03	Aplikasi dapat menampilkan daftar penyakit yang dapat diobati dengan tanaman obat tersebut.
FA-04	Pengguna dapat melihat resep atau tata cara dalam pembuatan ramuan obat untuk mengobati penyakit tersebut
FA-05	Pengguna dapat melakukan pencarian terhadap tanaman obat kemudian aplikasi akan menampilkan hasil pencarian.

4.2.4. Arsitektur Sistem

Sistem secara garis besar memiliki beberapa bagian. Pertama adalah aplikasi ensiklopedia tanaman obat sendiri yang

Gambar 4. 1 Arsitektur Sistem

Berikut ini merupakan penjelasan dari cara kerja sistem pada aplikasi ensiklopedia tanaman obat, serta dapat dilihat pada Gambar 4.1 di atas:

1. Pengguna mengakses halaman web ensiklopedia tanaman obat melalui web browser pada perangkatnya.
2. Web browser tersebut akan melakukan request HTTP melalui koneksi internet ke server dimana web ensiklopedia tersebut berada.
3. Web server yang terdapat pada server akan melakukan pengecekan terhadap perintah dari request tersebut yang kemudian akan menjalankan script PHP.
4. Ketika pengguna membuka halaman aplikasi dan ketika aplikasi melakukan query SPARQL terhadap RDF Store maka PHP SPARQL Lib akan menjembatannya. Jika query yang dilakukan tidak error maka OpenRDF Sesame mengirimkan data dengan format JSON sebagai output dari query.
5. Untuk informasi penyakit diambil dari DBpedia sehingga akan melakukan query ke DBpedia, dan DBpedia akan mengirimkan data yang diminta berbentuk JSON.
6. PHP SPARQL Lib akan melakukan parsing dari data JSON dan menambahkannya ke dalam kode HTML. Kemudian akan dikirimkan ke Web server.
7. Web server akan mengirimkan response ke client dalam bentuk HTML melalui protokol HTTP internet yang selanjutnya akan dikirimkan ke client.
8. Data yang berbentuk HTML tersebut yang telah sampai pada client akan diterjemahkan oleh web browser dan ditampilkan kepada pengguna berbentuk halaman web yang sesuai dengan permintaan pengguna.
9. OpenRDF Sesame juga berguna sebagai endpoint, agar ontologi tanaman obat dapat diakses serta dimanfaatkan oleh banyak orang. Pengguna dapat melakukan query SPARQL untuk mendapatkan data. OpenRDF Sesame

mengatur penanganan endpoint ontologi tersebut. Serta Pubby dapat diakses pengguna yang akan melakukan SPARQL query agar dapat memahami struktur ontologi tanaman obat.

Selanjutnya, diperlukan suatu pedoman untuk membangun aplikasi ini agar aplikasi ini dapat dikatakan telah menerapkan teknologi *semantic web* dan berbentuk ensiklopedia.

Untuk dapat dikatakan sudah menggunakan teknologi *semantic web* adalah harus sudah memenuhi syarat sebagai berikut:

1. Mendukung RDF, RDF-Schema dan juga mendukung OWL (Web Ontology Language).
2. Dukungan untuk SPARQL dan juga dukungan spesifikasi SPARQL Extension seperti select, hapus, tambah, dan update. Pada sistem ini hanya disediakan ekstensi select saja.
3. Dapat menyimpan RDF skala data besar secara efisien.
4. Memiliki kemampuan inferensi untuk ontologi OWL.
5. Dapat menyebarkan Linked Data.
6. Data mengenai tanaman obat dapat diakses dan digunakan oleh pihak luar dengan mengakses SPARQL Endpoint tanaman obat.

Sedangkan untuk dapat dikatakan sudah berbentuk ensiklopedia adalah harus sudah memenuhi syarat sebagai berikut:

1. Adanya artikel/topik, sub topik.
2. Adanya definisi artikel/topik dan diikuti penjelasan umum.
3. Adanya rujuk silang (cross reference) atau further more, see also.
4. Adanya paragraf, ilustrasi, gambar, grafik, table time line.
5. Tersusun dengan baik, contohnya disajikan secara sistematis alfabetis (A – Z), atau tematis, historis, kronologis.

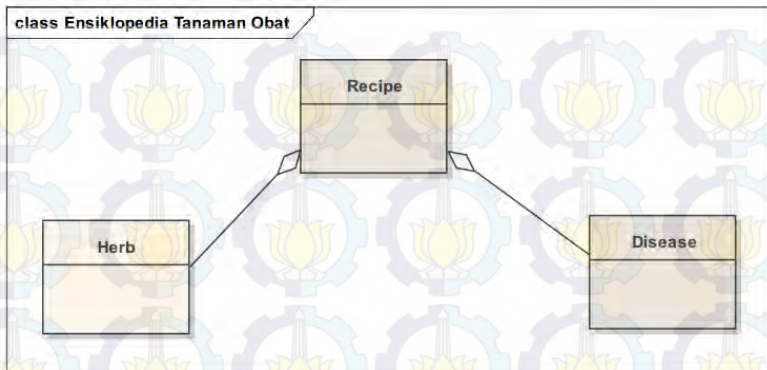
6. Adanya indeks.

Salah satu hal penting dalam bagian kebutuhan sistem adalah data tanaman obat itu sendiri yang akan dibuatkan ontologi. Pengumpulan data dilakukan terhadap tanaman obat. Dari studi pustaka dijelaskan bahwa dari sekian banyak jenis tanaman obat yang ada baru sekitar 300 jenis tanaman obat yang dimanfaatkan. Sehingga pengumpulan data ini adalah mengumpulkan informasi tanaman obat sebanyak 300 jenis. Untuk data tanaman obat penulis mengambil dari buku “Inventaris Tanaman Obat Indonesia (I)” jilid I dan II yang diterbitkan oleh Departemen Kesehatan dan Kesejahteraan Sosial RI pada tahun 2000 dan 2001.

4.3. Domain Model

Domain model merupakan pemodelan konseptual yang menampilkan objek-objek nyata dan relasi antar objek tersebut. Domain model digunakan untuk menyamakan istilah objek yang akan dipakai dalam pengerjaan aplikasi.

Recipe menjelaskan mengenai resep atau tata cara dalam mengolah tanaman obat menjadi obat untuk mengobati penyakit. Resep biasanya terdiri dari jenis tanaman serta bagian tanaman yang dapat digunakan obat, tata cara dalam pengolahan serta ditujukan untuk penyakit apa resep obat tersebut. Sehingga Recipe terdiri dari Disease dan Herb. Disease merupakan penjelasan untuk penyakit yang dapat diobati dengan tanaman obat. Yang terakhir Herb menjelaskan mengenai tanaman obat yang dapat digunakan obat untuk mengobati penyakit, dalam tanaman obat biasanya digunakan beberapa bagian dari tanaman obat tersebut, yang menjelaskan mengenai bagian dari tanaman obat yang dapat digunakan sebagai obat. Domain model pada pembuatan tanaman obat ini adalah seperti pada Gambar 4.2 di bawah ini.



Gambar 4. 2 Domain Model

4.4. Use Case Model

Use Case didapatkan dari hasil analisis terhadap fungsi aplikasi yang telah didefinisikan sebelumnya. Use Case merupakan model untuk menggambarkan kegiatan atau aktivitas yang dapat user lakukan dalam aplikasi ensiklopedia tanaman obat. Berdasarkan hasil studi literatur pada buku (Rosenberg & Stephens, 2007), penulis mengelompokkan use case ke dalam package-package. Pada model pengembangan perangkat lunak ICONIX, use case termasuk dalam tahapan analisis. Dalam menganalisis use case, penulis mencantumkan dari mana sumber use case berasal. Sumber use case berasal dari fungsi aplikasi, literatur, dan use case lain.

Use Case Melihat Daftar Tanaman Obat (UC-01) muncul karena fitur aplikasi dalam menampilkan daftar tanaman obat. Use Case Melihat Informasi Tanaman Obat (UC-02) muncul karena fitur dari aplikasi yang mengharuskan pengguna dapat melihat informasi mengenai tanaman obat. Use Case Melihat Daftar Penyakit (UC-03) muncul karena fitur aplikasi yang mengharuskan pengguna dapat melihat daftar penyakit yang dapat diobati dengan tanaman obat. Use Case Melihat Informasi Penyakit (UC-04) muncul

karena fungsi aplikasi yang dapat menampilkan informasi mengenai penyakit. Use Case Mencari Tanaman Obat (UC-05) muncul dari fitur aplikasi dalam menampilkan hasil pencarian tanaman obat yang diinginkan oleh pengguna. Untuk keseluruhan Use Case diadopsi dari referensi Tugas Akhir Masrizal Frisnanto, dengan perubahan domainnya saja. Tidak adanya penambahan Use Case dikarenakan dari hasil studi literatur sudah cukup lengkap. Pada Tabel 4.2 merupakan daftar Use Case.

Tabel 4. 2 Daftar Use Case

Kode	Use Case	Sumber
UC-01	Use Case Melihat Daftar Tanaman Obat	FA-01
UC-02	Use Case Melihat Informasi Tanaman Obat	FA-02, FA-04
UC-03	Use Case Melihat Daftar Penyakit	FA-03
UC-04	Use Case Melihat Informasi Penyakit	FA-03
UC-05	Use Case Mencari Tanaman Obat	FA-05

4.4.1. Use Case Diagram

Use Case diagram digunakan untuk menggambarkan hubungan use case dalam sistem dengan pihak luar sistem seperti pengguna. Hubungan antar use case dalam use case diagram dikelompokkan dalam package use case. Detail use case diagram tiap use case dijelaskan pada Lampiran A.

4.4.2. Deskripsi Use Case

Deskripsi use case merupakan pemaparan tentang hubungan antara user dan sistem. Deskripsi use case berisi nama use case,

kode use case, dan skenario. Detail deskripsi use case tiap use case dijelaskan pada Lampiran A.

4.5. GUI Prototype

GUI Prototype merupakan purwarupa antarmuka aplikasi ensiklopedia tanaman obat. GUI Prototype dibuat menggunakan tool yaitu aplikasi GUI Design Studio. GUI Prototype ini masih berupa desain sederhana dan akan digunakan sebagai panduan untuk pembuatan kode tampilan aplikasi. Selain itu, GUI Prototype digunakan untuk mempermudah visualisasi objek tampilan yang terdapat pada deskripsi use case.

Setiap GUI Prototype dimungkinkan dipakai dalam banyak use case. Oleh karena itu saat membuat GUI Prototype, penulis mencantumkan juga daftar use case yang menggunakan GUI Prototype tertentu. Pada saat pembuatan kode tampilan nanti dimungkinkan ada perubahan desain dari GUI Prototype yang telah dibuat. Berikut pada tabel 4. 3 Merupakan daftar GUI.

Tabel 4. 3 Daftar GUI Prototype

Kode	GUI Prototype	Use Case
GP-01	Halaman Pencarian	UC-05
GP-02	Halaman Daftar Tanaman Obat	UC-01
GP-03	Halaman Informasi Detail Tanaman Obat	UC-02
GP-04	Halaman Daftar Penyakit	UC-03
GP-05	Halaman Informasi Detail Penyakit	UC-04

Pada Lampiran B terlampir GUI pada sistem yang terdiri dari halaman atau view yang berbeda-beda.

4.6. Robustness Diagram

Robustness diagram merupakan gambaran objek dari use case. Robustness diagram menjembatani celah antara tahapan analisis dengan tahapan desain. Robustness diagram dibuat per use case, sehingga tiap use case memiliki satu robustness diagram. Robustness diagram aplikasi ensiklopedia tanaman obat dapat dilihat secara lengkap pada Lampiran C.

4.7. Sequence Diagram

Sequence diagram merupakan gambaran alur use case secara sistematis pada sistem. Sequence diagram bertujuan untuk menyempurnakan use case, dan mendetailkan class diagram. Sequence diagram dibuat per use case, sehingga tiap use case memiliki satu sequence diagram. Robustness diagram aplikasi ensiklopedia tanaman obat dapat dilihat secara lengkap pada Lampiran D.

4.8. Class Diagram

Class diagram merupakan gambaran class yang ada pada aplikasi. Class diagram bertujuan untuk memudahkan proses menulis kode aplikasi. Class diagram berisikan class, relasi antar class, atribut dan operasi tiap class. Class diagram erat hubungannya dengan use case. Class diagram dibentuk dari sequence diagram melalui tahapan desain pada ICONIX.

Tiap class pada Class Diagram muncul karena adanya kebutuhan untuk mengimplementasikan use case ke dalam kode. Class Recipe merupakan class yang berhubungan dengan Melihat Informasi Tanaman Obat (UC-02) dan Mencari Tanaman Obat (UC-05). Class Disease berhubungan dengan Melihat Daftar Penyakit (UC-03), Melihat Informasi Penyakit (UC-04), dan Mencari Tanaman Obat (UC-05). Class Herb berhubungan dengan use case Melihat Daftar Tanaman Obat (UC-01), Melihat Informasi Tanaman Obat (UC-02), Mencari Tanaman Obat (UC-05), Melihat Daftar Penyakit (UC-03), dan Melihat Informasi

Penyakit (UC-04). Tabel menjelaskan hubungan antara Class dan Use Case.

Tabel 4. 4 Relasi Antar Class dan Use Case

Class	Use Case
Recipe	UC-02, UC-05
Disease	UC-02, UC-03, UC-04, UC-05
Herb	UC-01, UC-02, UC-04, UC-05

Class diagram aplikasi ensiklopedia tanaman obat dapat dilihat secara lengkap pada Lampiran E.

4.9. Test Case

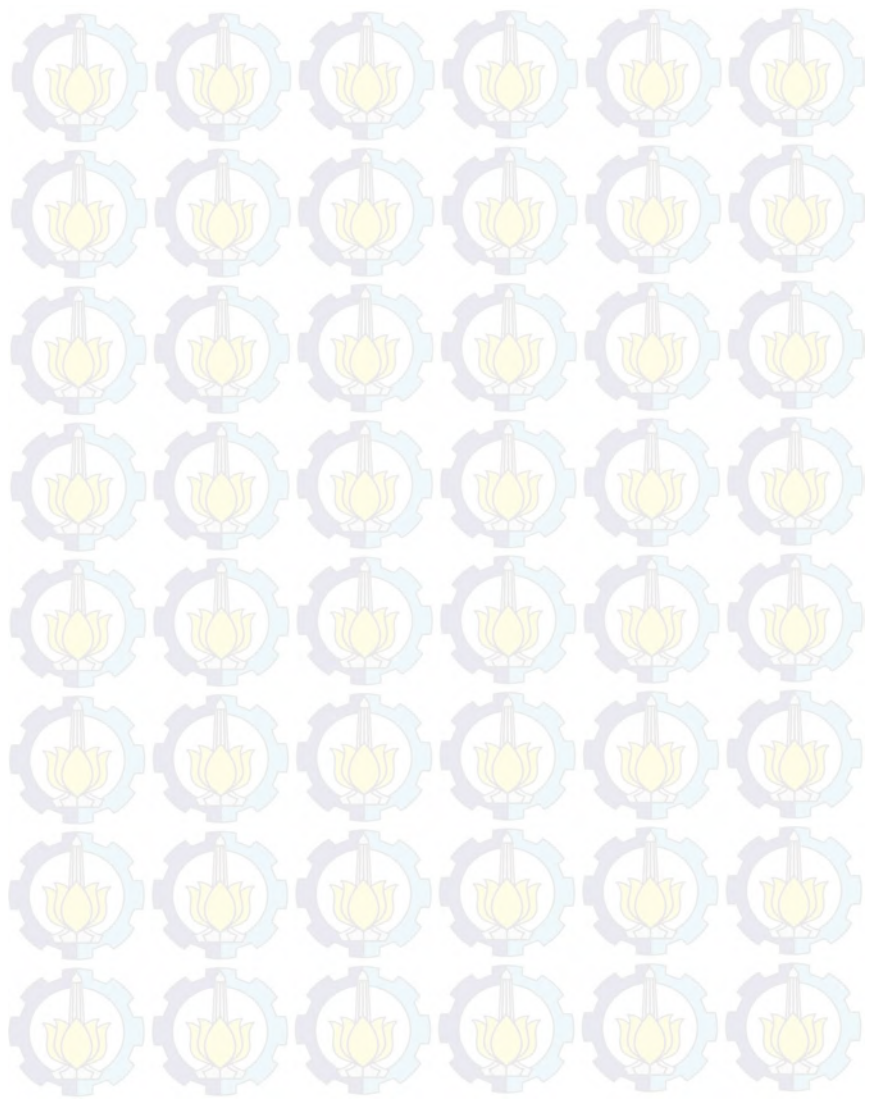
Test case merupakan skenario untuk uji coba aplikasi. Skenario dibuat berdasarkan skenario use case baik basic path maupun alternate path. Test case dijalankan setelah aplikasi sudah selesai dibuat. Test case dibuat dalam format tabel seperti pada Tabel 4. 11 dibawah.

Tabel 4. 5 Contoh Daftar Test Case

No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1					
2					
3					

Test case aplikasi ensiklopedia tanaman obat dapat dilihat pada Lampiran F.

(halaman ini sengaja dikosongkan)



BAB V

PEMBUATAN ONTOLOGI

Pada bagian ini dijelaskan mengenai pembuatan ontologi tanaman obat, sekaligus mengikuti dari metode pembuatan ontologi yaitu Methontology meliputi: Specification, Knowledge Acquisition, Conceptualisation, dan Integration, Implementation, Evaluation, dan Documentation.

5.1. Specification

Melakukan identifikasi terhadap tujuan-tujuan dari ontologi, serta kebutuhan-kebutuhan yang harus dipenuhi. Seperti pemilihan bahasa untuk ontologi. Pada pembuatan ontologi tanaman obat ini digunakan bahasa ontologi yaitu OWL serta dengan alat bantu untuk membuat ontologi yaitu Protégé. Pemilihan OWL dikarenakan kemampuannya dalam mempresentasikan hubungan yang ada dalam domain kemudian juga dikarenakan OWL menudukung beberapa fungsi yang tidak ada dalam bahasa ontologi lain seperti hubungan inversOf dan lain sebagainya. Nantinya dalam perancangan dan pengimplementasian dari ontologi menggunakan alat bantu Protégé, dipilih karena kemudahannya serta fitur yang disediakan dalam penyimpanan ontologi ke dalam XML/RDF.

5.2. Knowledge Acquisition

Melakukan pengumpulan pengetahuan mengenai domain, dalam hal ini adalah melakukan studi literatur terhadap tanaman obat.

5.3. Conceptualisation

Mendefinisikan kelas ontologi dan menyusun kelas tersebut dalam hirarki taksonomi dengan menggunakan proses pengembangan dari tingkat teratas ke tingkat terendah, mendefinisikan properti, mendefinisikan facets pada slot atau axiom pada properti. Pada bagian pendefinisian tanaman, penulis mengambil referensi dari sebuah website id.dbpedia.org untuk menjadi dasar dari ontologi tanaman obat. Pada website tersebut

dijelaskan mengenai vocabulary tanaman. Penulis dapat menyimpulkan ada beberapa vocabulary yang diperlukan untuk ontologi tanaman. Pada Gambar 5.1 berikut salah satu contoh dari Jahe yang memiliki URI <<http://id.dbpedia.org/page/Jahe>>:



Gambar 5. 1 Contoh URI Jahe pada id.dbpedia.org

Untuk penjelasana ilmiah mengenai tanaman ada beberapa vocabulari yang dijelaskan yaitu:

- [dbpprop-id:regnum](#) menjelaskan mengenai kingdom/kerajaan dari spesies tanaman.
- [dbpprop-id:ordo](#) menjelaskan mengenai ordo dari tanaman.
- [dbpprop-id: familia](#) menjelaskan mengenai keluarga atau famili dari tanaman.
- [dbpprop-id:genus](#) menjelaskan marga dari tanaman.
- [dbpprop-id:species](#) menjelaskan nama spesies dari tanaman.
- [dbpprop-id:binomial](#) menjelaskan mengenai nama ilmiah dari tanaman

Untuk tambahan vocabulari lain yang ada dan dianggap penting adalah [dbpedia-owl:abstract](#) atau [rdfs:comment](#) yang menjelaskan mengenai informasi singkat mengenai tanaman. Serta properti

tersebut yang digunakan untuk penyakit, menjelaskan singkat mengenai penyakit tersebut.

Selanjutnya untuk rancangan ontologi tanaman obat penulis mengambil referensi dari sebuah paper yang berjudul “Malaysia Indigenous Herbs Knowledge Representation”. Pada bagian dijelaskan mengenai pembangunan ontologi tanaman obat serta vocabularinya. Untuk kelas cara penggunaannya atau Preparation Method ada 6 sub class yaitu: Decoction, Drying, Extraction, Direct Usage, Peel dan Incised. Selanjutnya untuk manfaat dari tanaman yang dapat diambil sebagai obat yaitu: Leaf, Root, Stem, Bark, Fruit dan Whole Plant. Berikut ini merupakan Tabel 5.1 yang menjelaskan mengenai hubungan tanaman obat dengan manfaatnya.

Tabel 5. 1 Hubungan Tanaman Obat dengan Manfaatnya

No	Local Name	Medicinal Uses	Preparation Method	Preparation Category
1	Limau Purut	Dandruff	Squeeze	Extraction
2	Mengkudu	High Blood Pressure	Eaten Fresh	Direct Usage
		Cancer	Eaten Fresh	Direct Usage
		Skin Disease	Smearing	Direct Usage
3	Tea Tree	Wound	Crushed	Extraction

Pada tabel di atas dijelaskan mengenai representasi dari jenis tanaman obat yang dihubungkan dengan manfaatnya untuk obat penyakit tertentu serta cara untuk penggunaannya. Selanjutnya Tabel 5.2 berikut ini menjelaskan mengenai bagian dari tanaman yang dapat dimanfaatkan serta hubungannya dengan penyakit yang dapat diobati dengan bagian tanaman tersebut.

Tabel 5. 2 Hubungan Bagian Tanaman Obat untuk Penyakit

No	Local Name	Medicinal Uses	Plant Part	Preparation Method	Preparation Category
1	Limau Purut	Dandruff	Fruit	Squeeze	Extraction
2	Mengkudu	High Blood Pressure	Leaf	Eaten Fresh	Direct Usage
		Cancer	Fruit	Eaten Fresh	Direct Usage
		Skin Disease	Fruit	Smearing	Direct Usage
3	Tea Tree	Wound	Leaf	Crushed	Extraction

Sehingga untuk keseluruhan properti pada ontologi tanaman obat pada paper tersebut adalah seperti pada Tabel 5.3 sebagai berikut:

Tabel 5. 3 Daftar Property

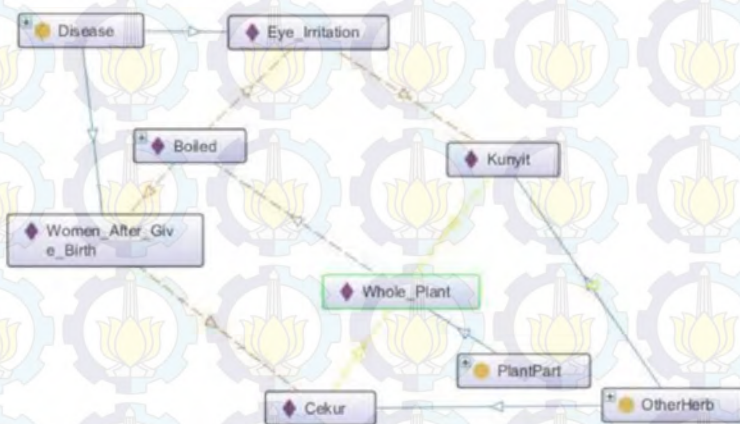
No	Property	Invers property
1	treatmentFor	treatedWith
2	instrumentFor	performedByInstrument
3	hasPart	isPartOf
4	processFor	usesProcess
5	includeSubprocess	subProcessOf

Berikut penjelasan dari masing-masing properti di atas:

1. treatedWith merupakan properti yang menjelaskan mengenai “diobati dengan”. Penyakit “treatedWith” Tanaman obat. Properti ini berlawanan dengan treatmentFor.
2. instrumentFor merupakan property yang menjelaskan “bagian untuk”. Bagian Tanaman “instrumentFor” Preparation Method. Properti ini berlawanan dengan performedByInstrument.
3. hasPart merupakan property “memiliki bagian”. Tanaman “part” daun/buah. Properti ini berlawanan dengan isPartOf.

4. processFor merupakan property yang menjelaskan “proses untuk”. Preparation method “processFor” penyakit. Properti ini berlawanan dengan usesProcess.
5. includeSubprocess merupakan properti yang menjelaskan “memiliki proses”. Preparation category “includeSubprocess” Preparation method. Properti ini berlawanan dengan subprocessOf.

Pada Gambar 5.2 ini merupakan contoh dari representasi ontologi tanaman obat yaitu kunyit:



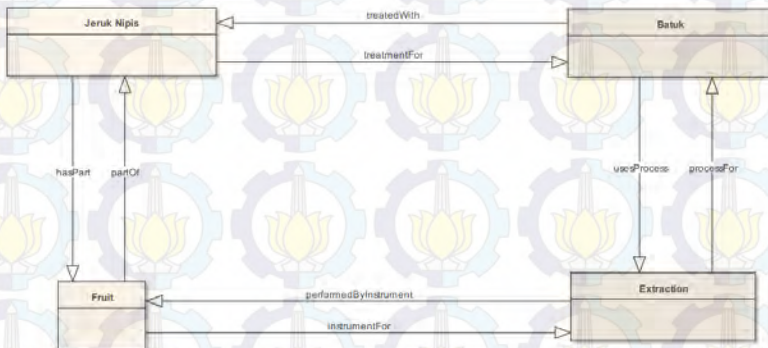
Gambar 5. 2 Contoh Representasi Tanaman Obat

Dari Gambar 5.2 di atas dapat diambil kesimpulan bahwa Kunyit dan Cekur merupakan jenis OtherHerb dan bagian tanaman yang dapat dimanfaatkan adalah semua bagian tanaman dengan cara direbus. Selanjutnya Kunyit dapat digunakan sebagai obat iritasi mata sedangkan Cekur dapat digunakan untuk obat bagi ibu setelah melahirkan anak.

5.4. Integration

Mempertimbangkan ontologi yang sudah ada atau berhubungan dengan domain tanaman obat, dengan menggunakan kembali vocabulary yang sudah ada.

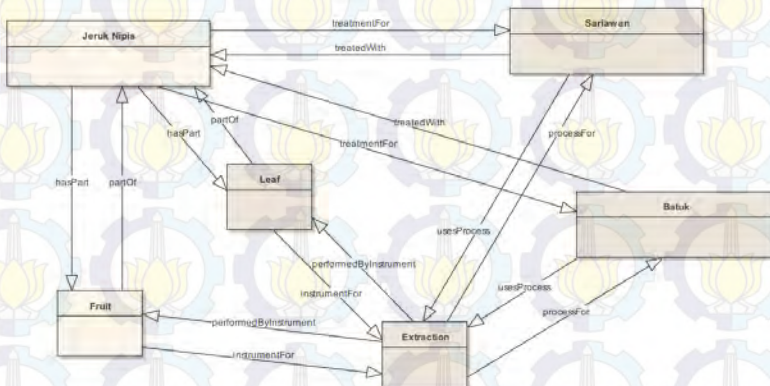
Penulis akan menggabungkan ontologi yang sudah ada baik dari referensi paper dan juga dari id.dbpedia.org. Untuk vocabulary penulis mengambil dari referensi paper sedangkan untuk keperluan aplikasi juga penulis mengambil detail informasi mengenai tanaman obat dari id.dbpedia.org. setiap resource diidentifikasi dengan URI yang membedakannya dengan resource yang lain sehingga diharapkan tidak ada resource yang memiliki URI yang sama misalkan untuk Jahe memiliki URI http://id.dbpedia.org/page/Jeruk_nipis. Jeruk nipis juga memiliki properti-properti mengenai kedudukannya dalam tanaman. Jadi nantinya penulis akan menggunakan URI dari tanaman untuk memasukkannya dalam ontologi tanaman obat yang akan dibangun. Untuk menghubungkan ontologi dengan yang dibuat digunakan properti owl:sameAs. Setiap dataset akan dihubungkan dengan dataset dari id.dbpedia.org. Untuk menghubungkan dengan penyakit dihubungkan dengan ontologi id.dbpedia.org.



Gambar 5. 3 Contoh hubungan Jeruk Nipis dengan Batuk

Penjelasan dari Gambar 5.3 di atas adalah Jeruk nipis yang buahnya diekstrak dapat digunakan untuk mengobati penyakit batuk. Sehingga Jeruk nipis dapat digunakan untuk mengobati Batuk.

Namun dari ontologi yang sudah ada menurut penulis kurang yaitu adanya keambiguan data yang menjelaskan hubungan antara bagian tanaman yang digunakan dengan Preparation Method dengan fungsi untuk obat penyakit yang berbeda sehingga tidak diketahui bagian dari tanaman yang mana yang dapat digunakan untuk obat antara penyakit tersebut. Untuk lebih jelasnya digambarkan pada Gambar 5.4 di bawah ini.

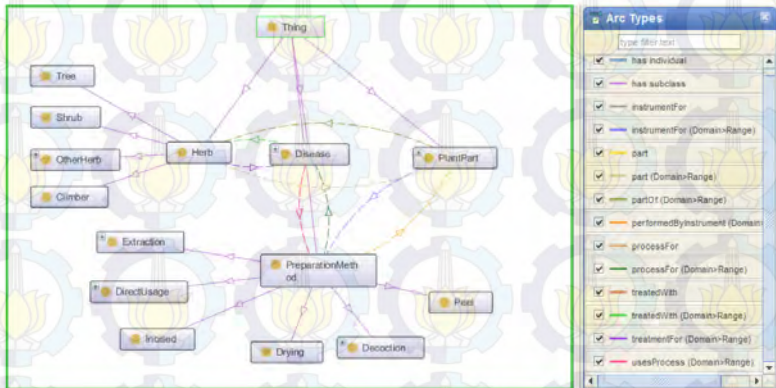


Gambar 5. 4 Pemisalan kondisi berbeda

Dari Gambar 5.4 dapat dilihat bahwa tidak diketahui manakah bagian dari Jeruk nipis yang dapat digunakan untuk obat Batuk daun atau buah. Serta untuk Sariawan bagian dari Jeruk nipis manakah yang dapat digunakan untuk obat, daun atau buah.

Ternyata juga masih belum adanya kelas yang menjelaskan mengenai resep atau penjelasan detail mengenai cara pengolahan bagian tanaman obat tersebut, sehingga penulis menambahkan bagian yang kurang tersebut. kelas tersebut adalah Recipe yang penulis ambil dari <http://schema.org/Recipe>. Recipe adalah

sekumpulan komposisi dengan cara penyajiannya untuk mempersiapkan makanan. Sedangkan dalam dunia kesehatan bisa dikatakan adalah sebuah cara pemakaian obat untuk penyakit tertentu. Sehingga untuk propertinya adalah isPartOf dan hasPart. Berikut ini representasi untuk kelas-kelas pada ontologi tanaman obat sebelum ada penambahan kelas Recipe.

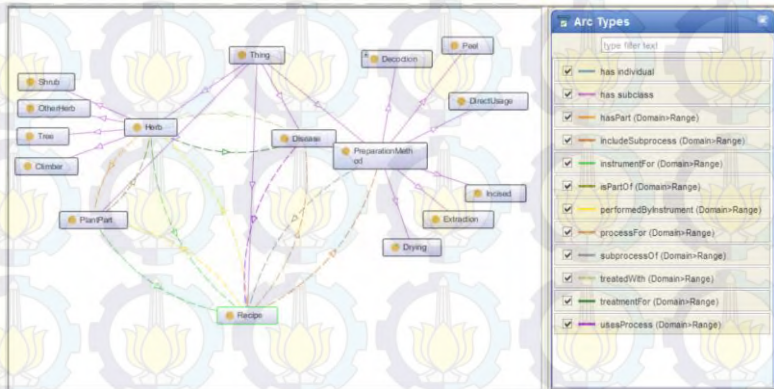


Gambar 5. 5 Gambaran ontologi sebelum penambahan kelas Recipe

Secara umum Gambar 5.5 di atas dapat dijabarkan sebagai berikut: Herb atau tumbuhan obat dibagi menjadi beberapa sub kelas yaitu: Tree, Shrub, Other Herb dan Climber. Pembagian ini berdasarkan jenis dari tanaman obat tersebut. kemudian tanaman obat tersebut dihubungkan dengan hubungan “treatedWith” Disease yang berarti tanaman obat tersebut digunakan untuk treatment penyakit. Selanjutnya pemanfaatan tanaman obat tersebut digunakan dari bagiannya sehingga ada kelas tersendiri yaitu PlantPart yang menjelaskan bagian tertentu dari tanaman obat yang digunakan untuk obat. Kemudian dari bagian tanaman obat tersebut dihubungkan dengan PreparationMethod menjelaskan mengenai cara pemanfaatan bagian tanaman obat tersebut. Untuk proses pemanfaatannya dibagi menjadi sub kelas: Extraction, DirectUsage, Incised, Drying, Decoction, Peel.

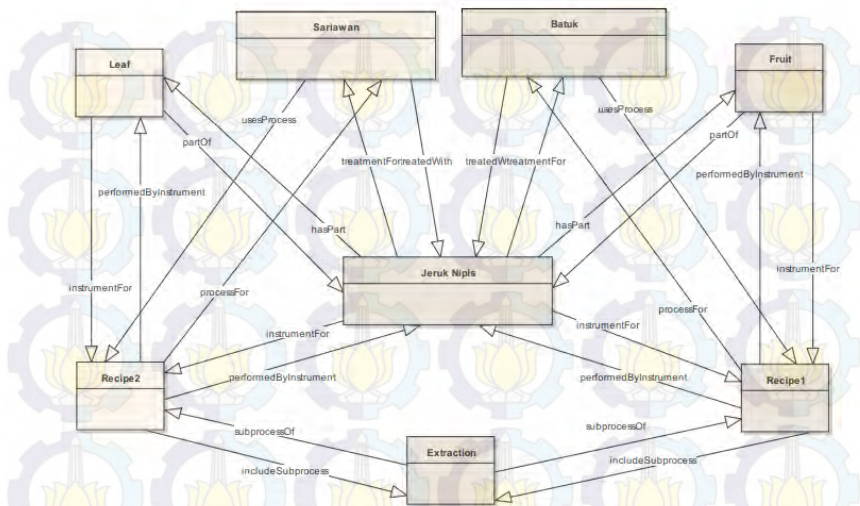
Pemabagian menjadi beberapa sub kelas ini menunjukkan dari cara pemanfaatan bagian tanaman obat tersebut untuk dijadikan obat.

Kemudian dengan penambahan kelas Recipe didapatkan rancangan ontologi sebagai berikut:



Gambar 5. 6 Gambaran ontologi setelah penambahan kelas Recipe

Dari Gambar 5.6 di atas dapat dilihat bahwa kelas Recipe menghubungkan antara Disease, PlantPart serta PreparationMethod. Dengan adanya kelas Recipe dapat dimungkinkan bagian dari tanaman obat memiliki fungsi sebagai obat dari penyakit yang berbeda walaupun dengan cara penyajian atau PreparationMethod yang sama. Pada Gambar 5.7 berikut contohnya:

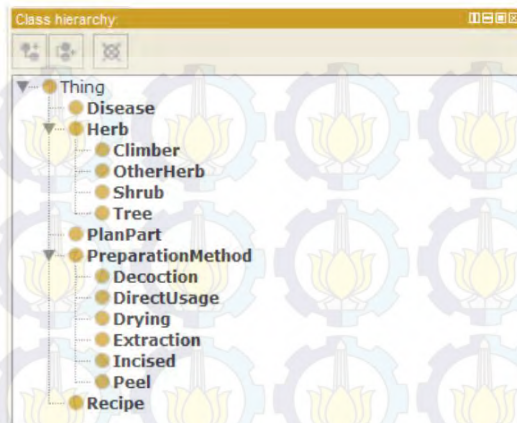


Gambar 5. 7 Contoh penerapan dengan Jeruk Nipis

Tahapan pada perancangan ontologi ini didapatkan beberapa definisi dari kelas-kelas yang akan digunakan serta property-property-nya.

5.4.1. Pendefinisian Kelas

Kelas merupakan salah satu komponen yang harus didefinisikan dalam ontologi. Dari hasil rancangan didapatkan beberapa kelas yaitu: Disease, Herb, PlantPart, PreparataionMethod, Recipe lebih jelasnya pada Gambar 5.8 berikut.



Gambar 5. 8 Kelas pada Ontologi Tanaman Obat

lebih jelasnya dijelaskan pada Tabel 5.4 di bawah ini.

Tabel 5. 4 Penjelasan Kelas pada Ontologi Tanaman Obat

No.	Nama Kelas	Penjelasan
1	Disease	Merupakan kelas yang menjelaskan mengenai penyakit.
2	Herb	Merupakan kelas yang menjelaskan mengenai tanaman obat. Dari kelas Herb ini dibagi beberapa sub kelas yang menjelaskan kategori dari tanaman obat, yaitu: Climber , Shrub , Tree , OtherHerb .
3	PlantPart	Merupakan kelas yang menjelaskan mengenai bagian tanaman obat yang dapat digunakan sebagai obat.

4	PreparationMethod	Menjelaskan kelas yang menjelaskan mengenai cara yang digunakan untuk memproses tanaman obat tersebut sebagai obat. Kelas PreparationMethod ini dibagi menjadi beberapa sub kelas yang menjelaskan kategori dari cara pemrosesan. Sub kelasnya yaitu: Decoction, DirectUsage, Drying, Extraction, Incised, Peel.
5	Recipe	Merupakan kelas yang menjelaskan mengenai resep tata cara pengolahan tanaman obat sebagai obat. Diambil dari < http://schema.org/Recipe >

5.4.2. Pendefinisian Properti

Properti merupakan hal lain yang penting dalam membangun ontologi. Properti inilah yang akan menjelaskan hubungan antara kelas dan objek. Ada dua tipe properti yaitu object properties dan datatype properties. Object properties menjelaskan hubungan antara objek-objek sedangkan datatype properties menjelaskan komponen dari tiap objek tersebut. Penulis dalam membuat rancangan ontologi ini berpedoman pada ontologi yang sudah ada. Oleh karena itu penulis menggunakan prefix dari URI berikut:

1. rdfs dari <<http://www.w3.org/2000/01/rdfschema#>> .
2. foaf dari <<http://xmlns.com/foaf/0.1/>> .
3. owl dari <<http://www.w3.org/2002/07/owl#>> .
4. rdf dari <<http://www.w3.org/1999/02/22-rdfsyntax-ns#>> .
5. dbpedia-owl <<http://dbpedia.org/ontology/>> .
6. schema dari <<http://schema.org/>> .
7. botany dari <<http://ontologi.es/biol/botany#>> .
8. osspr dari <<http://data.ordnancesurvey.co.uk/ontology/spatialrelations/>> .

9. biol dari < <http://purl.org/NET/biol/ns#>> . Sedangkan untuk beberapa properti maupun kelas yang belum ada penulis membuat sendiri oleh karena itu penulis mendefinisikan prefix untuk ontologi yang nantinya akan penulis buat. Penulis menggunakan prefix iih, yang merupakan singkatan untuk tanaman obat Indonesia.

Tabel 5.5 ini merupakan object properties dalam ontologi tanaman obat:

Tabel 5. 5 Daftar Object Property

No	Nama Properti	Invers	Penjelasan
1	hasPart	isPartOf	Menjelaskan hubungan bagian objek dari kelas yang satu dengan objek dari kelas yang lain.
2	treatmentFor	treatedWith	Menjelaskan hubungan kegunaan dari tanaman obat terhadap penyakit.

3	instrumentFor	performedByInstrument	Merupakan hubungan bagian dari resep terhadap tanaman obat.
4	processFor	usesProcess	Merupakan hubungan antara metode penggunaan dengan penyakit.
5	includeSubprocess	subprocessOf	Merupakan hubungan antara resep dengan metode penggunaan.
6	foaf:depiction	-	Properti untuk URL gambar dari suatu objek.

Tabel 5.6 ini merupakan data properties dalam ontologi tanaman obat:

Tabel 5. 6 Daftar Data Property

NO	Nama Property	Penjelasan
1	foaf:name	Merupakan properti untuk menjelaskan nama dari objek,

		properti ini untuk
		menjelaskan nama umum dari objek. Digunakan untuk menyatakan nama dagang atau umum dari tanaman obat.
2	foaf:givenName	Digunakan untuk menyatakan nama lain dari suatu objek, dapat digunakan untuk menyatakan nama daerah dari tanaman obat atau pun nama penyakit.
3	owl:sameAs	Merupakan properti untuk menjelaskan hubungan yang sama dengan objek yang sudah ada.
4	rdfs:label	Merupakan properti untuk menjelaskan label dari objek.
5	dbpedia-owl:abstract	Merupakan properti untuk menjelaskan keterangan singkat mengenai objek.
6	rdfs:comment	Merupakan properti untuk menjelaskan bagian dari objek tersebut. Biasanya digunakan untuk menjelaskan kelas juga.
7	biol:name	Properti untuk menjelaskan nama ilmiah dari tanaman obat

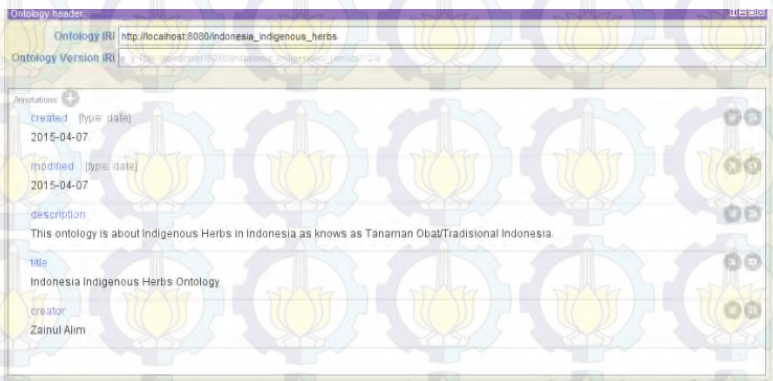
8	biol:commonName	Merupakan properti untuk menjelaskan nama dari objek, properti ini untuk menjelaskan nama umum dari objek. Digunakan untuk menyatakan nama dagang atau umum dari
		tanaman obat atau penyakit.
9	botany:division	Properti untuk menjelaskan divisi dari tanaman obat.
10	botany:subdivision	Properti untuk menjelaskan sub divisi dari tanaman obat.
11	botany:class	Properti untuk menjelaskan kelas dari tanaman obat.
12	botany:order	Properti untuk menjelaskan bangsa dari tanaman obat.
13	botany:family	Properti untuk menjelaskan suku dari tanaman obat.
14	botany:genus	Properti untuk menjelaskan marga dari tanaman obat.
15	botany:species	Properti untuk menjelaskan jenis dari tanaman obat.
16	osspr:contains	Properti untuk menjelaskan kandungan yang dimiliki oleh tanaman obat.
17	schema:recipeInstructions	Properti untuk menjelaskan langkah-langkah dalam penyajian resep dari tanaman obat.

5.5. Implementation

Merancang konsep ontologi dan merepresentasikannya ke dalam bahasa ontologi yang dipilih, dalam hal ini akan membuat instance jenis-jenis tanaman obat dan penyakit serta mengisi nilai slot pada instance.

5.5.1. Pendefinisian Ontologi

Ontologi yang dibangun penulis memberi nama dengan Indonesia Indigenous Herbs yang mengartikan bahwa ontologi yang penulis buat adalah ontologi mengenai tanaman obat yang ada di Indonesia. URI ontologi yang penulis buat adalah http://localhost:8080/indonesia_indigenous_herbs. Sedangkan untuk prefix yang penulis gunakan adalah iih yang merupakan kependekan dari nama ontologi dapat dilihat pada Gambar 5.9.



Gambar 5. 9 Pendefinisian Nama Ontologi

ini daftar prefix yang digunakan:

Tabel 5. 7 Daftar Prefix

No	Prefix	URI
1	Rdfs	http://www.w3.org/2000/01/rdf-schema#

2	Foaf	http://xmlns.com/foaf/0.1/
3	Owl	http://www.w3.org/2002/07/owl#
4	Rdf	http://www.w3.org/1999/02/22-rdf-syntax#
5	dbpediaowl	http://dbpedia.org/ontology/
6	schema	http://schema.org/
7	botany	http://ontologi.es/biol/botany#
8	osspr	http://data.ordnancesurvey.co.uk/ontology/spatialrelations/
9	dc	http://purl.org/dc/elements/1.1/
10	dcterms	http://purl.org/dc/terms/
11	iih	http://localhost:8080/indonesia_indigenous_herbs
12	xsd	http://www.w3.org/2001/XMLSchema#
13	biol	http://purl.org/NET/biol/ns#

Prefix tersebut dimasukkan ke dalam Protege dapat dilihat pada Gambar 5.10.

Prefix	Value
biol	http://localhost:8080/indonesia_indigenous_herbs/
botany	http://purl.org/NET/biol/ns#
dbpedia-owl	http://ontologi.es/biol/botany#
dc	http://dbpedia.org/ontology/
dcterms	http://purl.org/dc/elements/1.1/
foaf	http://purl.org/dc/terms/
iih	http://xmlns.com/foaf/0.1/
osspr	http://localhost:8080/indonesia_indigenous_herbs/
owl	http://data.ordnancesurvey.co.uk/ontology/spatialrelations/
rdf	http://www.w3.org/2002/07/owl#
rdfs	http://www.w3.org/1999/02/22-rdf-syntax-ns#
schema	http://www.w3.org/2000/01/rdf-schema#
xsd	http://schema.org/
	http://www.w3.org/2001/XMLSchema#

Gambar 5. 10 Memasukkan Prefix

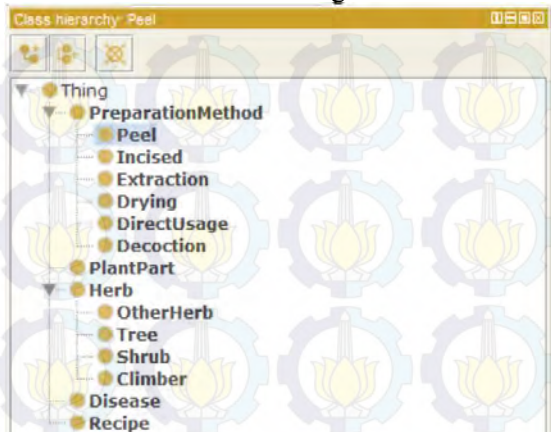
5.5.2. Pembuatan Kelas

Pada bagian pembuatan kelas mengacu pada Tabel 5.8 kelas berikut:

Tabel 5. 8 Daftar Kelas dan Sub Kelas

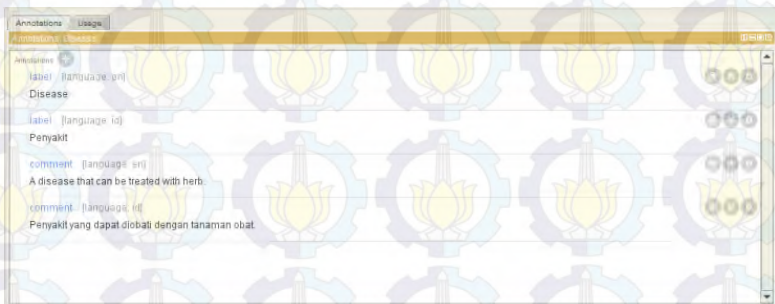
No.	Nama Kelas	Sub Kelas	Prefix dan URI
1	Disease	-	iih <http://localhost:8080/in donesia_indigenous_her bs>
2	Herb	Climber, Shrub, Tree, OtherHerb	iih <http://localhost:8080/in donesia_indigenous_her bs>
3	PlantPart	-	iih <http://localhost:8080/in donesia_indigenous_her bs>
4	PreparationMethod	Decoction, DirectUsage, Drying, Extraction, Incised, Peel.	iih <http://localhost:8080/in donesia_indigenous_her bs>
5	Recipe	-	schema
			<http://schema.org/Reci pe>

Berikut setelah dimasukkan ke Protege:



Gambar 5. 11 Daftar Kelas pada Protege

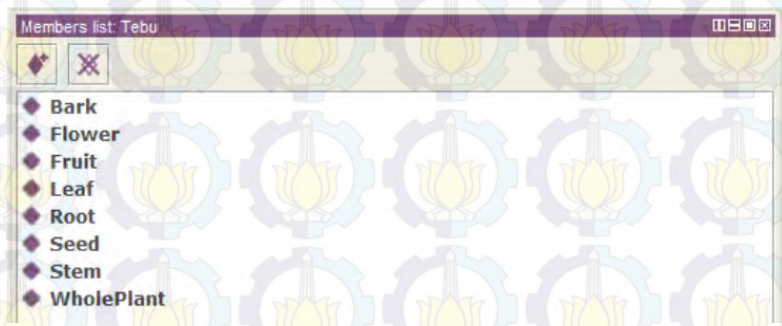
Selanjutnya diberikan anotasi untuk memberikan penjelasan singkat mengenai kelas dan sub kelas tersebut. Ini dapat juga dilakukan pada properti agar kelas dan properti tersebut mudah dipahami maksud dan artinya. Berikut contoh pemberian anotasi pada kelas Disease. Dapat dilihat pada Gambar 5.12 di bawah ini.



Gambar 5. 12 Anotasi Kelas

Dari Gambar 5.12 di atas dapat dilihat untuk kelas Disease diberikan anotasi label dan comment yang memberikan penjelasan singkat mengenai kelas Disease tersebut.

Selanjutnya adalah membuat instance dari bagian tanaman atau PlantPart serta PreparationMethod. Untuk PlantPart sendiri berpedoman pada sumber acuan sehingga untuk instance PlantPart ada: Leaf, Root, Stem, Bark, Fruit, WholePlant. Namun dikarenakan ada beberapa bagian lain yang biasa digunakan sehingga penulis perlu menambahkan Flower. Dapat dilihat pada Gambar 5.13 di bawah ini.



Gambar 5. 13 Instance pada Kelas PlantPart

Namun untuk penambahan instance PreparationMethod akan dilakukan bersamaan dengan pemasukan data tanaman obat pada langkah selanjutnya.

5.5.3. Pembuatan Object Property

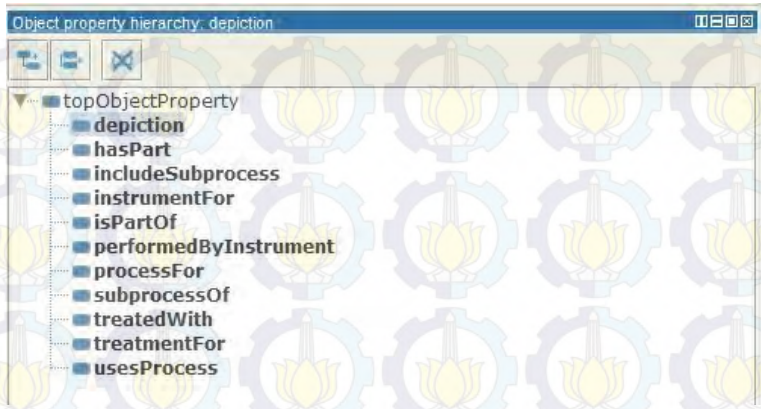
Tabel 5.9 berikut merupakan object property:

Tabel 5. 9 Daftar Object Property

No.	Nama Properti	Prefix dan URI
1	hasPart	dcterms < http://purl.org/dc/terms/ >

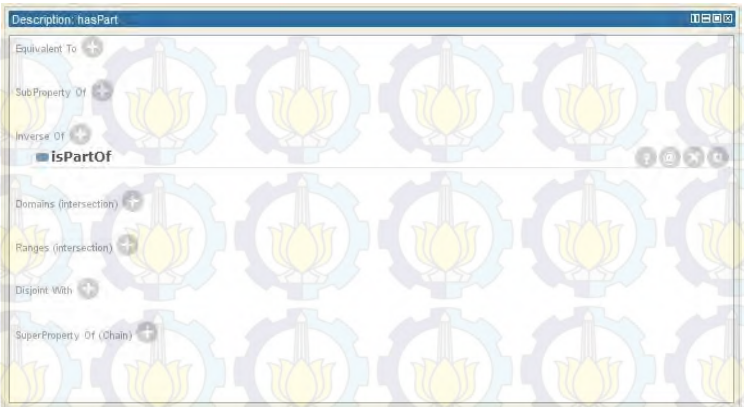
2	treatmentFor	iih < http://localhost:8080/indonesia_indigenous_herbs >
3	instrumentFor	iih < http://localhost:8080/indonesia_indigenous_herbs >
4	processFor	iih < http://localhost:8080/indonesia_indigenous_herbs >
5	includeSubprocess	iih < http://localhost:8080/indonesia_indigenous_herbs >
6	isPartOf	dcterms < http://purl.org/dc/terms/ >
7	treatedWith	iih < http://localhost:8080/indonesia_indigenous_herbs >
8	performedByInstrument	iih < http://localhost:8080/indonesia_indigenous_herbs >
9	usesProcess	iih < http://localhost:8080/indonesia_indigenous_herbs >
10	subprocessOf	iih < http://localhost:8080/indonesia_indigenous_herbs >
11	depiction	Foaf < http://xmlns.com/foaf/0.1/ >

Dari tabel di atas kemudian dimasukkan ke dalam Protege. Dapat dilihat pada Gambar 5.14 di bawah ini.



Gambar 5. 14 Memasukkan Object Property

Sekaligus penjelasan dari invers dan hubungan setiap object property dengan kelas. Invers menyatakan kebalikan dari suatu properti sehingga dapat dipahami bahwa salah properti tersebut bermakna berlawanan dengan properti yang lain. Berikut contoh penerapan properti *hasPart* yang bermakna berlawanan dengan properti *isPartOf* tampak pada Gambar 5.15 di bawah ini.



Gambar 5. 15 Pendefinisian Invers

Selanjutnya diberikan juga anotasi label dan comment dari masing-masing properti. Berikut salah satu pemberian anotasi pada properti treatedWith dapat dilihat pada Gambar 5.16 di bawah ini.



Gambar 5. 16 Anotasi Object Property

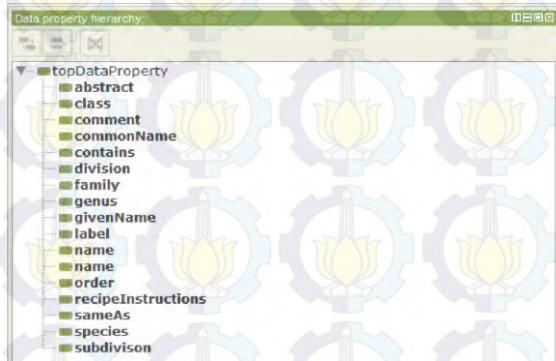
5.5.4. Pembuatan Data Property

Tabel 5.10 berikut merupakan data properti:

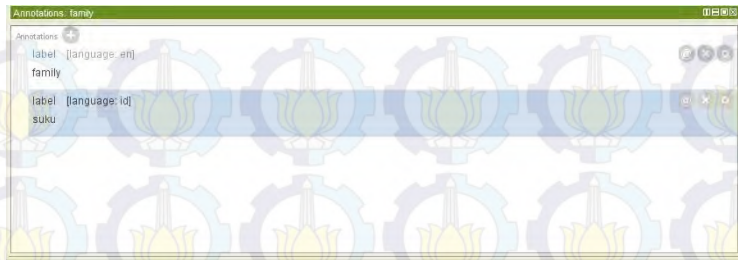
Tabel 5. 10 Daftar Data Property

No.	Prefix dan Data Property	No.	Prefix dan Data Property
1	foaf:name	11	botany:class
2	foaf:givenName	12	botany:order
3	owl:sameAs	13	botany:family
4	rdfs:label	14	botany:genus
5	dbpedia-owl:abstract	15	botany:species
6	rdfs:comment	16	osspr:contains
7	biol:name	17	schema:recipeInstructions
8	biol:commonName		
9	botany:division		
10	botany:subdivision		

Dari Tabel 5.10 di atas dimasukkan ke dalam Protege dapat dilihat pada Gambar 5.17 di bawah ini.

**Gambar 5. 17 Memasukkan Data Property**

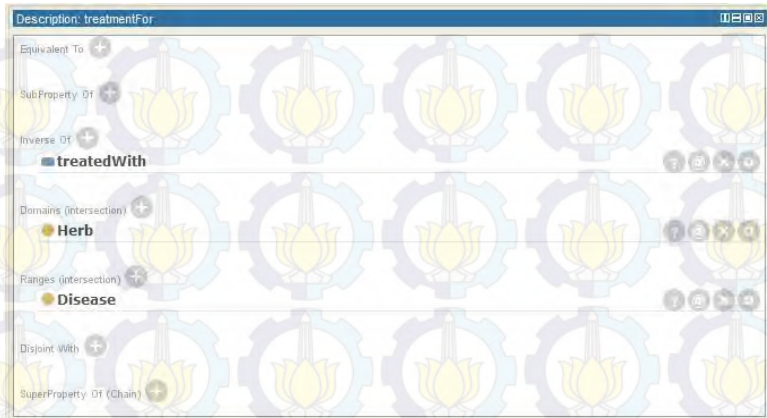
Kemudian diberikan anotasi dan penjelasan mengenai properti tersebut. Untuk lebih jelasnya dapat dilihat contoh data properti family pada Gambar 5.18 di bawah ini.



Gambar 5. 18 Anotasi Data Property

5.5.5. Pendefinisian Relasi Antar Kelas dan Properti

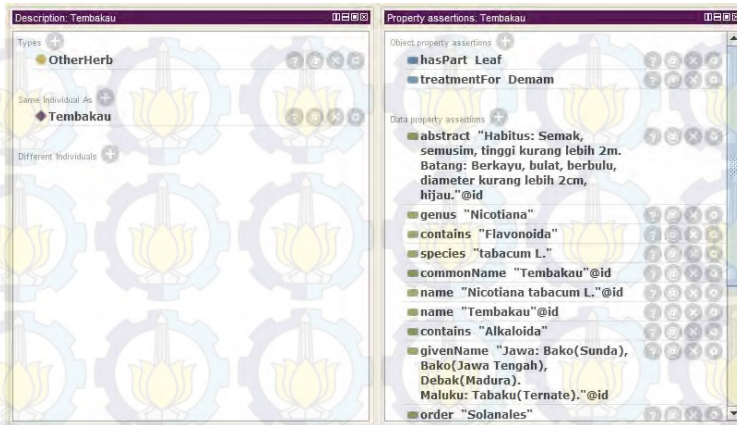
Langkah selanjutnya adalah memberikan relasi antar kelas dengan object property. Contohnya adalah ada dua kelas Herb dan Disease. Relasi yang menghubungkan adalah “treatmentFor”. Contoh pernyataan yang ada adalah tanaman obat dapat digunakan untuk obat penyakit. Sehingga dari dua kelas tersebut terbentuk relasi Herb “treatmentFor” Disease. Dalam ontologi ada dua istilah Domain dan Range. Domain merupakan fungsi/kelas/objek asal yang dihubungkan dengan properti, sedangkan Range merupakan fungsi/kelas/objek tujuan. Bisa disimpulkan Domain adalah subjek, properti adalah predikat, dan Range adalah objek. Kalimat pernyataan sederhana di atas terdiri dari frase SPO, sehingga untuk Domain adalah Herb dan Range adalah Disease. Berikut lebih jelasnya dalam implementasi di Protege pada Gambar 5.19 di bawah ini:



Gambar 5. 19 Pendefinisian Relasi

5.5.6. Pemasukan Data Tanaman Obat

Tahap ini merupakan tahap yang membutuhkan waktu yang lama dikarenakan data yang dimasukkan secara manual serta dengan jumlah data yang banyak. Sesuai dengan rencana awal, data yang akan dimasukkan sebanyak 300 jenis tanaman obat. Data yang akan dimasukkan meliputi nama, serta taksonomi dari tanaman obat, kemudian penjelasan singkat mengenai tanaman obat baik ciri-ciri, kandungan kimia dan manfaat dari tanaman obat serta tata cara pengolahan tanaman obat sebagai obat untuk penyakit. Berikut salah satu contoh memasukkan data tanaman obat yaitu Tembakau pada Gambar 5.20 di bawah ini:



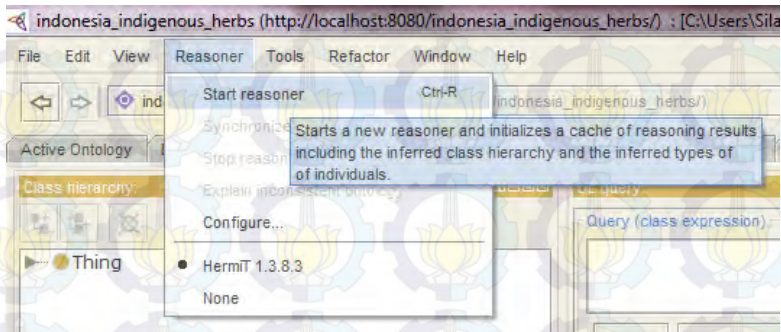
Gambar 5. 20 Memasukkan Instance Tanaman Obat

5.6. Evaluation

Melakukan verifikasi serta validasi dari ontologi dengan memeriksa konsistensi logika, sintaks, dan semantik dari elemen-elemen yang ada pada ontologi. Pemeriksaan konsistensi dapat mencakup klasifikasi otomatis yang mendefinisikan konsep-konsep baru berdasarkan properti dan relasi [17].

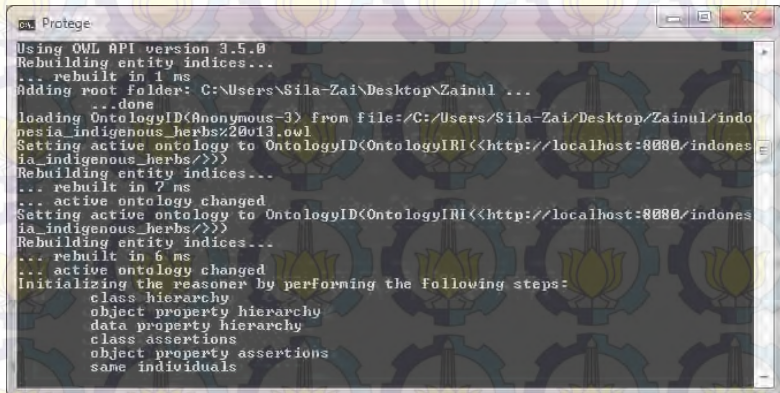
5.6.1. Pemeriksaan Konsistensi

Tahapan pemeriksaan konsistensi dengan menggunakan plugin HermiT pada Protégé. Langkahnya adalah memilih HermiT pada menu Reasoner serta menjalankannya.



Gambar 5. 21 Memilih dan menjalankan Reasoner Hermit

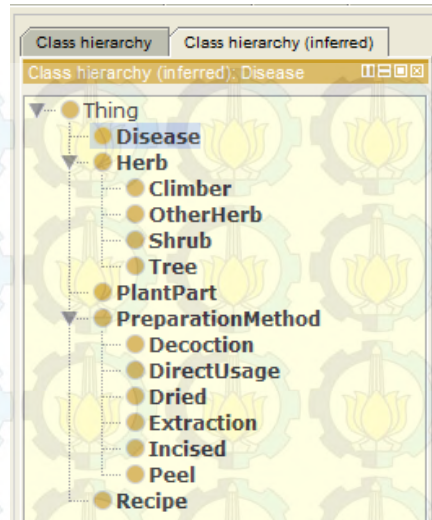
Kemudian proses reasoning akan berjalan dapat dilihat pada command line, seperti pada Gambar 5.22 di bawah ini.



Gambar 5. 22 Proses me-load data

Jika proses loading selesai maka pada tab Classess dan pada Class Hierarchy.

Pada Gambar 2.23 di bawah ini menunjukkan proses loading selesai.



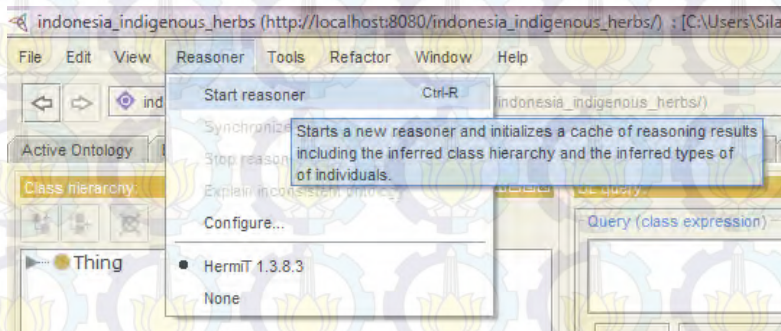
Gambar 5. 23 Hasil reasoning HermiT

Dengan melakukan reasoning dengan HermiT, kelas yang inkonsisten akan terlihat merah dan dikenal sebagai *Probe Class* yang ditandai dengan warna merah. Sehingga apabila pada model ontology yang dibuat terdapat kelas yang inkonsisten, kelas tersebut akan menjadi berwarna merah. Pada uji coba kali ini, Protégé tidak menunjukkan warna kelas yang merah dan tidak ada penjelasan kelas yang inkonsisten, lihat pada Gambar 5.23 di atas.

5.6.2. Pemeriksaan Relasi

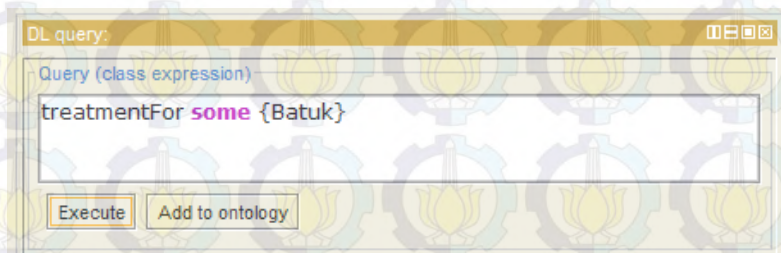
Tahapan pemeriksaan untuk melihat relasi antar kelas dan objek adalah dengan menggunakan plugin OntoGraf pada Protégé, Hasil dari pemeriksaan ini ditampilkan dengan visualisasi, jadi hubungan apa saja yang berhubungan dengan yang mau dilihat. Misalnya untuk melihat relasi yang terjalin dengan instance “Batuk”. Langkahnya masuk pada tab Ontograf, pada kolom “Search” masukkan kata kunci “Batuk”. Kemudian klik tombol “Search”.

Langkahnya pada menu Reasoner, pilih HermiT dan Start Reasoner.



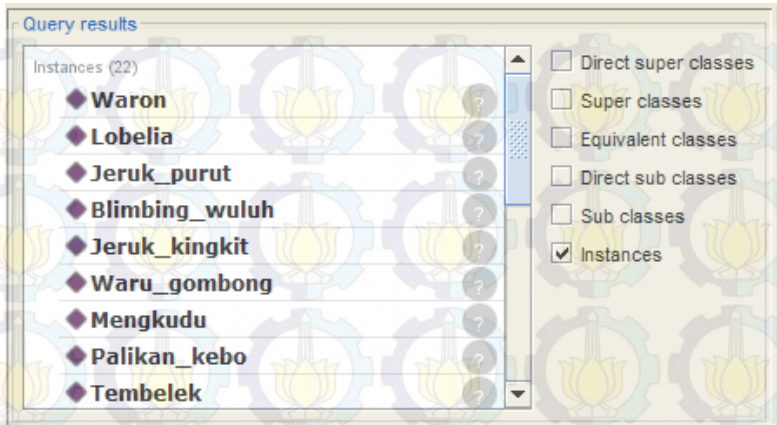
Gambar 5. 26 Menjalankan Reasoner

Selanjutnya pada bagian tab DL-Query ketikkan query seperti pada Gambar 5.27 di bawah ini.



Gambar 5. 27 DL-Query untuk melihat tanaman obat untuk obat Batuk

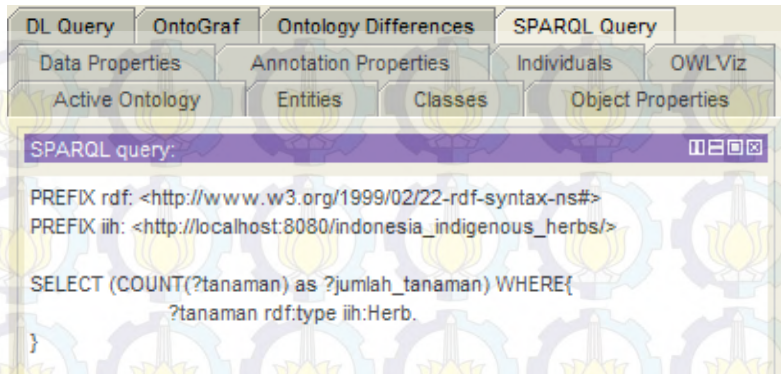
Jika query pada Gambar 5.27 di atas dijalankan, maka hasilnya akan seperti pada Gambar 5.28 di bawah ini yang merupakan tanaman obat yang digunakan sebagai obat Batuk. Hasilnya menunjukkan sebanyak 22 tanaman obat yang dapat digunakan untuk obat penyakit Batuk.



Gambar 5. 28 Hasil tanaman obat untuk obat penyakit "Batuk"

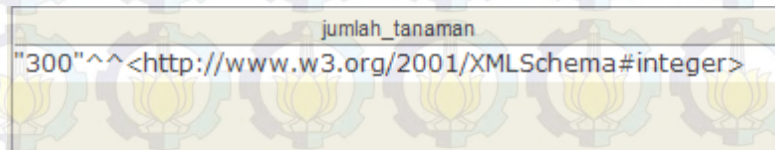
5.6.4. Pemeriksaan Query SPARQL

Pemeriksaan query SPARQL dilakukan untuk mengecek apakah ontology sudah dapat diquery dengan SPARQL. Langkah ini hampir sama dengan langkah pada DL-Query yang membedakan hanyalah bahasa query-nya. Query terhadap ontologi yang sudah di simpan melalui pugin SPARQL Query pada Protege. Jumlah tanaman obat yang dibuat dalam ontologi ini adalah sebanyak 300, oleh karena itu perlu dilakukan pengecekan apakah jumlah tanaman obat tersebut sudah sebanyak 300.



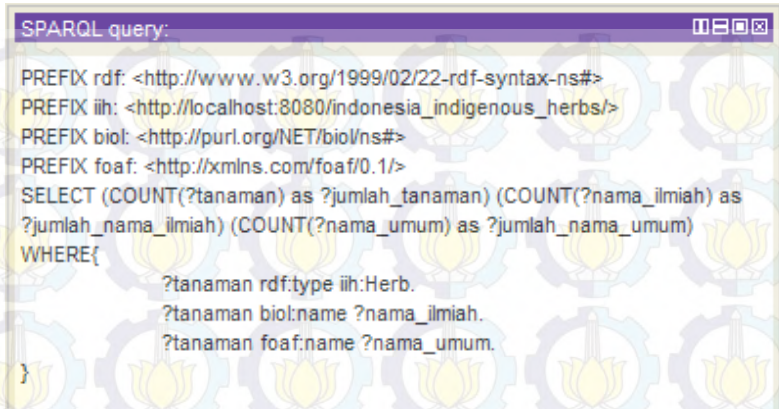
Gambar 5. 29 Query SPARQL untuk menghitung jumlah tanaman obat.

Setelah query pada Gambar 5.29 di atas dijalankan dan hasilnya adalah seperti pada Gambar 5.30 di bawah ini.



Gambar 5. 30 Jumlah Tanaman Obat

Selanjutnya dilakukan pengecekan apakah dari tiap tanaman obat tersebut telah memiliki propertinya masing-masing. Berikut query SPARQL pada Gambar 5.31 untuk mengeceknya.



```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX iih: <http://localhost:8080/indonesia_indigenous_herbs/>
PREFIX biol: <http://purl.org/NET/biol/ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT (COUNT(?tanaman) as ?jumlah_tanaman) (COUNT(?nama_ilmiah) as
?jumlah_nama_ilmiah) (COUNT(?nama_umum) as ?jumlah_nama_umum)
WHERE{
    ?tanaman rdf:type iih:Herb.
    ?tanaman biol:name ?nama_ilmiah.
    ?tanaman foaf:name ?nama_umum.
}

```

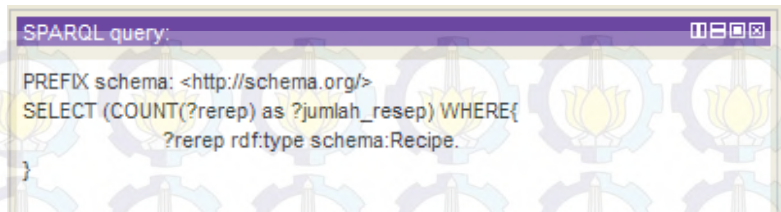
Gambar 5. 31 SPARQL query untuk mengecek properti

Pada query di atas untuk mengecek apakah setiap tanaman obat sudah memiliki properti nama umum dan nama ilmiahnya. Setelah query di atas dijalankan hasilnya seperti pada Gambar 5.32 di bawah ini.

jumlah_tanaman	jumlah_nama_ilmiah	jumlah_nama_umum
"300"^^<http://www	"300"^^<http://www	"300"^^<http://www

Gambar 5. 32 Hasil jumlah untuk tiap properti

Selanjutnya adalah melakukan pengecekan terhadap jumlah resep atau tata cara dalam pengolahan tanaman obat. Pada ontologi yang sudah dibuat terdapat 300 resep sama seperti jumlah tanaman obat. Setiap tanaman obat memiliki satu resep. Berikut query pada Gambar 5.33 yang dijalankan.



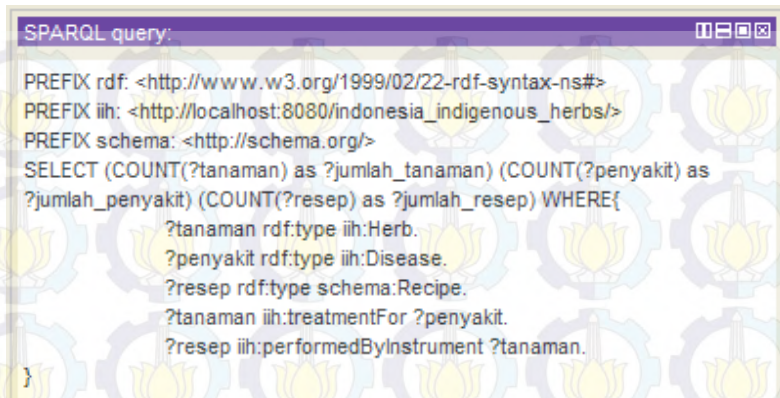
Gambar 5. 33 Query untuk menghitung jumlah resep

Setelah query dijalankan didapatkan hasil seperti pada Gambar 5.34 di bawah ini.



Gambar 5. 34 Hasil jumlah resep

Pengecekan selanjutnya adalah melihat hubungan tanaman obat dan penyakit, dalam hal ini properti yang digunakan adalah `treatmentFor`, dari ontologi yang dibuat harusnya tiap tanaman obat memiliki hubungan `treatmentFor` dengan satu penyakit. Serta dengan menggunakan satu resep. Untuk mengeceknya dengan query seperti pada Gambar 5.35 berikut ini, dan seharusnya jumlahnya semuanya ada 300.



```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX iih: <http://localhost:8080/indonesia_indigenous_herbs/>
PREFIX schema: <http://schema.org/>
SELECT (COUNT(?tanaman) as ?jumlah_tanaman) (COUNT(?penyakit) as
?jumlah_penyakit) (COUNT(?resep) as ?jumlah_resep) WHERE{
    ?tanaman rdf:type iih:Herb.
    ?penyakit rdf:type iih:Disease.
    ?resep rdf:type schema:Recipe.
    ?tanaman iih:treatmentFor ?penyakit.
    ?resep iih:performedByInstrument ?tanaman.
}

```

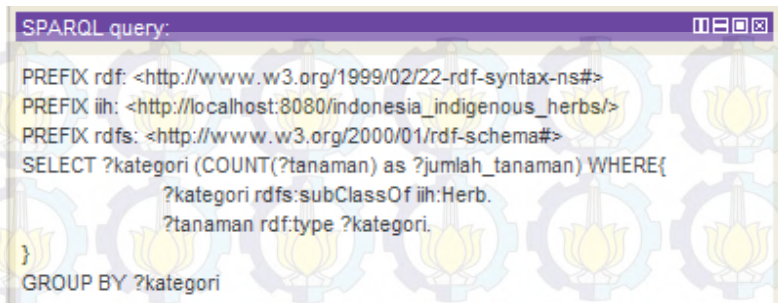
Gambar 5. 35 Query untuk melihat jumlah tanaman, penyakit, dan resep

Menghasilkan hasil seperti pada Gambar 5.36 di bawah ini.

jumlah_tanaman	jumlah_penyakit	jumlah_resep
"300"^^<http://ww	"300"^^<http://ww	"300"^^<http://ww

Gambar 5. 36 Hasil jumlah tanaman, penyakit dan resep

Selanjutnya adalah melakukan pengecekan jumlah tanaman obat berdasarkan kategori dari tanaman obat tersebut, dan jika ditotal semuanya tetap berjumlah 300. Berikut adalah query pada Gambar 5.37 untuk mengeceknya.



```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX iih: <http://localhost:8080/indonesia_indigenous_herbs/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?kategori (COUNT(?tanaman) as ?jumlah_tanaman) WHERE{
    ?kategori rdfs:subClassOf iih:Herb.
    ?tanaman rdf:type ?kategori.
}
GROUP BY ?kategori
  
```

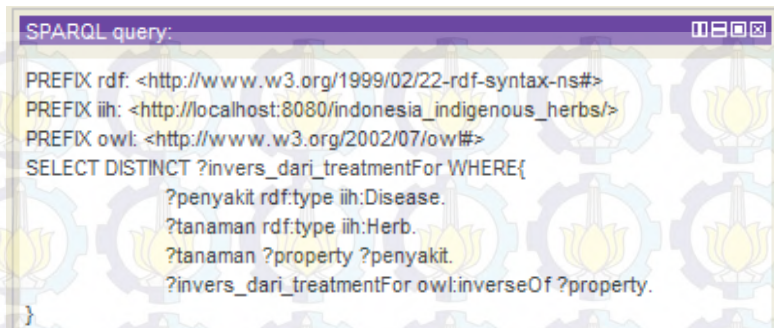
Gambar 5. 37 Query untuk menghitung jumlah tanaman tiap kategori

Hasilnya seperti Gambar 5.38 di bawah ini. Jika setiap kategori dijumlahkan maka totalnya sebanyak 300.

kategori	jumlah_tanaman
Tree	"92"^^<http://www.w3.org/2
Climber	"19"^^<http://www.w3.org/2
Shrub	"188"^^<http://www.w3.org/
OtherHerb	"1"^^<http://www.w3.org/20

Gambar 5. 38 Hasil jumlah tanaman obat tiap kategori

Bagian terakhir adalah mengecek tiap properti yang bersifat berlawanan dengan dengan properti yang lain. Misalnya treatmentFor berlawanan dengan treatedWith, oleh karena itu akan dicek apakah fungsi invers tersebut sudah benar. Berikut query pada Gambar 5.39 yang akan dilakukan, dan seharusnya menghasilkan treatedWith.



Gambar 5. 39 Query untuk melihat invers dari treatmentFor

Hasilnya adalah pada Gambar 5.40 di bawah ini.

invers_dari_treatmentFor
treatedWith

Gambar 5. 40 Hasil invers dari treatmentFor

Setelah melakukan pengecekan terhadap ontologi dengan cara yang sudah dilakukan di atas dengan memeriksa konsistensi sintaks serta relasi dapat disimpulkan bahwa ontologi yang sudah dibangun sudah valid dan konsisten baik dari segi logika dan sintaksnya. Penulis mengambil data mengenai tanaman obat ini dari buku Inventaris Tanaman Obat Indonesia (I) Jilid I dan II sehingga untuk data tanaman obat tersebut dapat dipertanggung jawabkan, dalam Tugas Akhir ini penulis membuat ontologi dari data tanaman obat tersebut.

5.7. Documentation

Melakukan pemeriksaan kembali dari setiap langkah yang dilakukan dan memastikan langkah yang dilakukan telah selesai. Langkah-langkah yang telah dilakukan dalam membangun

ontologi tanaman obat adalah seperti Tabel 5.11 di bawah ini dengan tanda checklist (✓) menandakan bahwa langkah tersebut telah dilalui, dan tanda silang (×) menandakan bahwa langkah tersebut belum selesai.

Tabel 5. 11 Dokumentasi Kegiatan

No.	Langkah	Keterangan
1	Specification	✓
2	Knowledge acquisition	✓
3	Conceptualisation	✓
4	Integration:	✓
	Pendefinisian Kelas	✓
	Pendefinisian Properti	✓
5	Implementation:	✓
	Pendefinisian Ontologi	✓
	Pembuatan Kelas	✓
	Pembuatan Object Property	✓
	Pembuatan Data Property	✓
	Pendefinisian Relasi Antar Kelas dan Property	✓
	Pemasukan Data Tanaman Obat	✓
6	Evaluation:	✓
	Pemeriksaan konsistensi logika dan sintaks	✓
7	Documentation	✓

✓ : Selesai.

× : Belum selesai.

BAB VI

IMPELEMENTASI DAN UJI COBA

Pada bab ini dijelaskan proses pada tahapan implementasi dan uji coba Aplikasi Ensiklopedia Tanaman Obat serta pembuatan Ontologi Tanaman Obat Indonesia. Adapun yang dibahas adalah lingkungan implementasi, pembuatan ontologi, konfigurasi, pembuatan aplikasi, dan uji coba.

6.1. Lingkungan Implementasi

Pengembangan aplikasi dan ontologi tanaman obat ini menggunakan komputer dengan spesifikasi hardware dan software pada Tabel 6.1.

Tabel 6. 1 Spesifikasi Komputer

Komputer Klien	
Prosesor	Intel(R) Core(TM) i3 M380 2.53 GHz
Memori	5 GB RAM
Sistem Operasi	Windows 8 Pro 64 bit
Komputer Server	
Prosesor	Intel(R) Core(TM) i5-3330 3.00 GHz
Memori	2 GB RAM
Sistem Operasi	Windows 7 Pro 32 bit

Proses pengembangan aplikasi ini menggunakan beberapa teknologi terkait dengan bahasa pemrograman, database, editor, library, dan server yang digunakan yang dapat dilihat pada Tabel 6.2 berikut.

Tabel 6. 2 Software Pendukung Lain yang Digunakan

Webserver	Apache 2.2.21, Tomcat 7.0.16
Bahasa Pemrograman	PHP 5.3.8
Editor	Notepad++

Browser	Google Chrome 41.0.2272.118 m
Framework	• OpenRDF Sesame • Protege • JDK 7u40 x64
Library	• Pubby • SPARQL Lib

6.2. Konfigurasi

Bagian ini menjelaskan tentang konfigurasi sistem sebelum memulai penulisan kode program. Adapun konfigurasi yang dibahas adalah OpenRDF Sesame, Direktori Aplikasi, Arsitektur Aplikasi, dan Konfigurasi Aplikasi.

6.2.1. OpenRDF Sesame

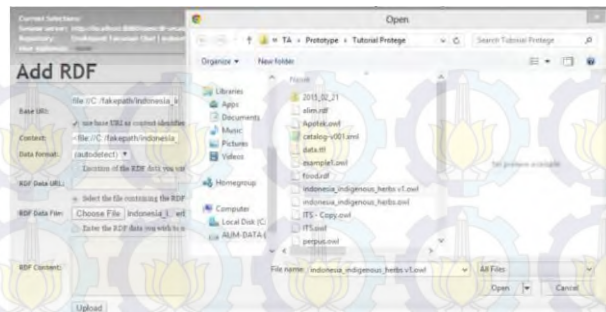
OpenRDF Sesame yang sudah diunduh kemudian dan diatur sesuai petunjuknya. Setelah itu masuk ke dalam pengaturan OpenRDF dan membuat *Repository* baru.

Lihat contoh pada Gambar 6.1 di bawah ini.



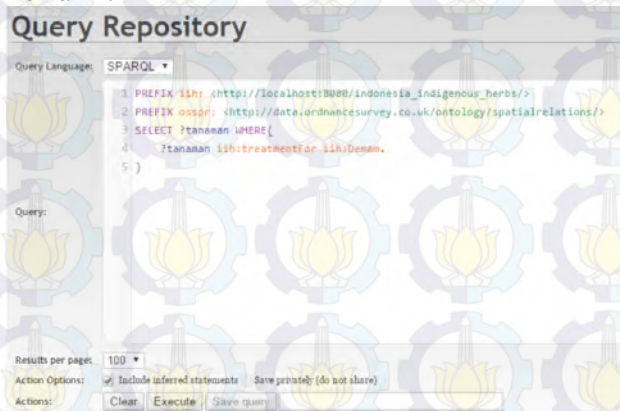
Gambar 6.1 Mengisi Informasi Repository

Setelah membuat Repository baru selanjutnya mengunggah file *.owl. Lihat pada Gambar 6.2 di bawah ini untuk lebih jelasnya.



Gambar 6. 2 Mengunggah File Ontologi *.owl

Untuk memastikan bahwa ontologi yang ditambahkan telah benar dan dapat diakse atau di-query, maka akan dipastikan dengan melakukan query SPARQL sederhana. Query di bawah ini untuk mengambil tanaman obat yang dapat digunakan untuk obat penyakit “Demam”.



Gambar 6. 3 Menjalankan Query

Hasilnya akan seperti pada Gambar 6.4 di bawah ini.



Gambar 6. 4 Hasil Query

Setelah berhasil dapat dipastikan nanti ontologi tersebut dapat di-query oleh aplikasi yang nantinya akan dibuat. Namun jika gagal atau tidak mendapatkan hasil bisa jadi query yang salah atau ontologinya yang salah oleh karena itu dibutuhkan penyelesaian masalah tersebut sebelum menginjak pada tahapan pembuatan aplikasi.

6.2.2. Pubby

Pubby merupakan salah satu framework yang digunakan untuk menyajikan ontologi dalam bentuk front-end linked data. Fungsi utama dari Pubby adalah untuk memudahkan pengguna dalam mengeksplor dan melihat ontologi yang sudah dipublikasikan. Pada folder WEB-INF, file config.ttl diedit dengan file editor (Notepad++) dan diganti menjadi kode seperti pada Gambar 6.5 di bawah ini.

```

1  @prefix   conf:
2  <http://richard.cvmaniak.de/2007/pubby/config.rdf#> .
3  @prefix meta: <http://example.org/metadata#> .
4  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5  .
6  @prefix rdfa: <http://www.w3.org/2000/01/rdfa-schema#> .
7  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
8  @prefix owl: <http://www.w3.org/2002/07/owl#> .
9  @prefix dc: <http://purl.org/dc/elements/1.1/> .
10 @prefix dcterms: <http://purl.org/dc/terms/> .
11 @prefix iih:
12 <http://localhost:8080/indonesia_indigenous_herbs/> .
13
14 # Server configuration section
15 <> a conf:Configuration;
16
17     # Project name for display in page titles
18     conf:projectName "Indonesia Indigenous Herbs (IIH)";
19
20     # Homepage with description of the project for the link in the page header
21     <http://localhost:8080/indonesia_indigenous_herbs/> conf:projectHomepage
22
23     # The Pubby root, where the webapp is running inside the servlet container.
24     <http://localhost:8080/indonesia_indigenous_herbs/> conf:webBase
25
26
27
28     conf:dataset [
29         # SPARQL endpoint URL of the dataset
30         conf:spaqqlEndpoint <http://localhost:8080/openrdfesame/repositories/indonesia_indigenous_herbs/>
31
32         # Default graph name to query (not necessary for most endpoints)
33         conf:spaqqlDefaultGraph <http://dmedia.org/>
34
35         # Common URI prefix of all resource URIs in the
36         SPARQL dataset
37         conf:datasetBase
38         <http://localhost:8080/indonesia_indigenous_herbs/>
39     ];

```

Gambar 6. 5 Kode config.ttl

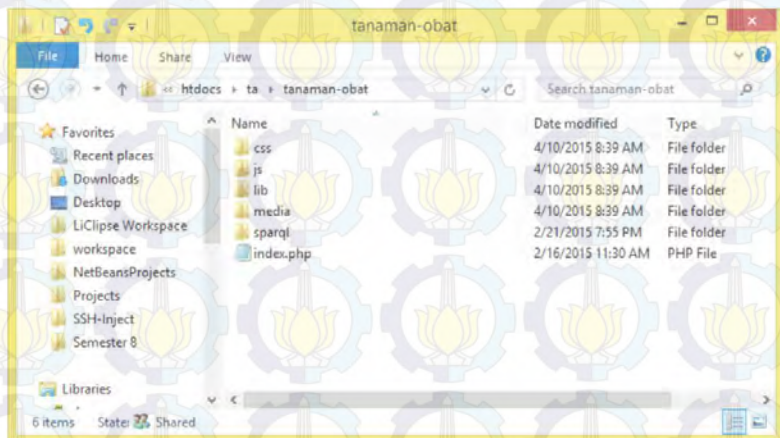
Setiap instance atau individual dari tanaman obat atau yang lainnya memiliki URI sendiri, oleh karena itu jika ingin melihat struktur dari setiap instance maka pada alamat yang telah disebutkan di atas ditambahkan dengan nama individual atau instance. Contohnya jika ingin melihat data Randu, maka dapat mengakses alamat http://localhost:8080/indonesia_indigenous_herbs/page/Randu.



Gambar 6. 6 Salah Satu Halaman untuk Instance Randu

6.2.3. Direktori Aplikasi

Pada direktori aplikasi tanaman obat terdiri dari 5 folder utama. Folder css berisi file-file untuk kode tampilan dari halaman web aplikasi tanaman obat. Dapat dilihat pada Gambar 6.7 di bawah ini.



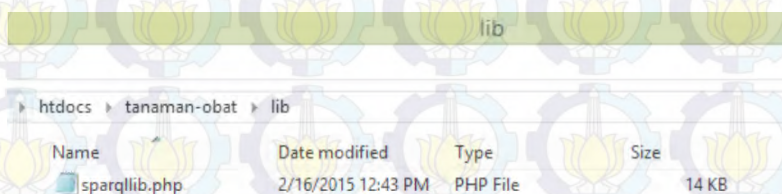
Gambar 6. 7 Direktori Aplikasi

Folder js berisi file-file yang memuat kode javascript yang dibutuhkan dalam aplikasi. Sedangkan folder lib berisi file-file yang memuat kode dari library yang digunakan untuk aplikasi khususnya yang diperlukan dalam menangani SPARQL. Yang terakhir adalah folder sparql merupakan direktori untuk halaman SPARQL Endpoint yang dimaksud untuk mendemonstrasikan query SPARQL dari ontologi tanaman obat yang sudah dipublikasikan lewat OpenRDF Sesame.

6.2.4. Konfigurasi Aplikasi

sparqllib.php

Telah dijelaskan pada bagian sebelumnya, aplikasi ini menggunakan SPARQL library dalam penanganan query SPARQL dalam PHP. File library terdapat di dalam folder lib pada root direktori.



Gambar 6. 8 File library SPARQL PHP-Lib

Pada penggunaan file library tersebut digunakan seperti dalam penggunaan pada Gambar 6.9 di bawah ini:

```

require_once( "lib/sparql.php" );
$db = sparql_connect(
"http://localhost:8080/openrdf-sesame/repositories/
indonesia_indigenous_herbs" );
if( !$db ) { print sparql_errno() . ": " .
sparql_error() . "\n"; exit; }
sparql_ns( "foaf", "http://xmlns.com/foaf/0.1/" );

```

Gambar 6. 9 Penggunaan library

Pada kode di atas sparql endpoint yang digunakan adalah endpoint dari alamat ontologi yang sudah dipublish pada OpenRDF Sesame. Pada bagian sparql_ns menjelaskan mengenai name space atau prefix yang digunakan. Prefix “foaf” digunakan untuk URI “http://xmlns.com/foaf/0.1/”.

Bagian selanjutnya adalah query SPARQL yang akan digunakan. Contoh query yang digunakan untuk mengambil variabel “nama” dan “nama_ilmiah” adalah seperti pada Gambar 6.10 di bawah ini:

```

$spqrql = "
PREFIX iih: <http://localhost:8080/indonesia_indigenous_herbs/>
PREFIX biol: <http://purl.org/NET/biol/ns#>

SELECT ?nama ?nama_ilmiah WHERE{
  ?s rdf:type iih:Herb.
  ?s foaf:name ?nama.
  ?s biol:name ?nama_ilmiah.
  filter (STRSTARTS(?nama, ".$abjad.")) .
}
ORDER BY ?nama
";
$result = sparql_query( $spqrql );

```

Gambar 6. 10 Contoh Query SPARQL

Hasil dari query tersebut akan menghasilkan data dalam array. Setelah itu mengambil data yang diinginkan.

model.php

File ini merupakan file yang berisi kelas-kelas serta method yang digunakan dalam aplikasi. Secara umum dalam file model.php terdapat 3 kelas yaitu: **Herb**, **Disease**, dan **Recipe**.

```
<?php

class Herb
{
    _____

class Disease
{
    _____

class Recipe
{
    _____
?>
```

Gambar 6. 11 Kelas pada model.php

Kelas **Herb** merupakan kelas yang meliputi method yang menangani pengembalian data untuk tanaman obat. Ada 4 function atau method yaitu: **getAllHerbs**, merupakan method yang mengembalikan semua data tanaman obat. **getHerbByAlphabet**, merupakan method yang mengembalikan data tanaman obat yang huruf depan dari nama tanaman obat tersebut sesuai dengan abjad yang diinginkan. **getHerbByURI**, merupakan method yang mengembalikan data tanaman obat yang sesuai dengan nama tanaman obat yang diinginkan. **findHerb**, merupakan method yang mengembalikan data tanaman obat yang memiliki kriteria dengan data yang ingin dicari, baik yang sesuai dengan nama, nama daerah, kandungan dan lain sebagainya.

```

class Herb
{
    function getAllHerbs ()
    {

    }

    function getHerbByURI ($suri)
    {

    }

    function getHerbByAlphabet ($abjad)
    {

    }

    function findHerb($kata_kunci)
    {

    }
}

```

Gambar 6. 12 Method pada Kelas Herb

Kelas **Disease** merupakan kelas yang meliputi method yang menangani pengembalian data untuk penyakit. Ada satu function atau method yaitu: **getDiseaseByURI** merupakan method yang mengambil data dari ontologi tanaman obat dan dari id.dbpedia.org, **getHerbByAlphabet** merupakan method untuk mengambil data tanaman obat yang difilter berdasarkan abjad, **getDiseaseByName**, merupakan method yang mengembalikan data penyakit penyakit tertentu. **getDiseaseByHerb**, merupakan method yang mengembalikan penyakit yang dapat diobati dengan tanaman obat yang diinginkan.

```

class Disease
{
    function getDiseaseByURI($suri)
    {

    }

    function getDiseaseByAlphabet ($abjad)
    {

    }

    function getDiseaseByName ($disease)
    {

    }

    function getDiseaseByHerb($herb)
    {

    }
}

```

Gambar 6. 13 Method pada Kelas Disease

Kelas **Recipe** merupakan kelas yang meliputi method yang menangani pengembalian data untuk resep. Ada 2 function atau method yaitu: **getRecipe**, merupakan method yang mengembalikan data resep atau tata cara pembuatan ramuan obat yang menggunakan tanaman tertentu sebagai obat dan digunakan untuk obat penyakit tertentu.

```
function getRecipe ($herb, $disease)
{
```

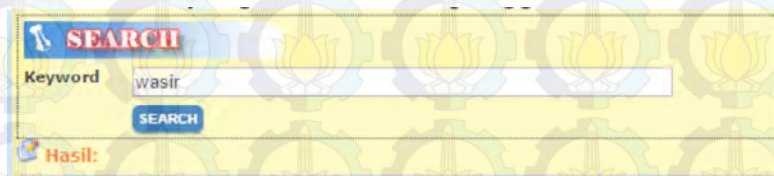
Gambar 6. 14 Method pada Kelas Recipe

6.3. Pembuatan Aplikasi

Bagian ini menjelaskan tentang penerapan desain yang telah dibuat dengan cara penulisan kode program. Adapun yang akan dibahas adalah mengenai Menu utama dengan fitur pencarian tanaman obat dan Menu daftar tanaman obat.

6.3.1. Menu Utama

Menu Utama merupakan menu halaman pertama pada saat aplikasi ensiklopedia tanaman obat di buka. Pada bagian desain dijelaskan bahwa pada menu utama ini terdapat fitur pencarian tanaman obat, pencarian tanaman obat ini mengacu pada kata kunci yang diisikan oleh pengguna.



Gambar 6. 15 Fitur Pencarian

Pada saat pengguna menekan tombol “Search”, maka aplikasi akan menjalankan method **showHasilPencarian(kata_kunci)** yang memakai bahasa pemrograman jQuery dan Javascript, tujuan dalam penggunaan jQuery agar aplikasi tanpa perlu mengalami loading, serta juga agar tampilan halaman terlihat statis dan tidak berubah-ubah. Dari fungsi di atas akan menampilkan halaman pencarian dimana halaman pencarian tersebut akan memanggil method **findHerb(\$kata_kunci)** pada kelas **Herb**. Namun sebelumnya akan dilakukan pengecekan terlebih dahulu apakah kata kunci sudah diisi oleh pengguna apa belum. Jika kata kunci telah diisi maka method **findHerb** akan dijalankan.

```
$a = new Herb;
$hasil = $a -> findHerb($kata_kunci);
...
```

Gambar 6. 16 Memanggil Method findHerb

Pada method **findHerb** tersebut, akan dilakukan query SPARQL terhadap ontologi dan dicocokkan dengan kriteria kata kunci yang dimasukkan oleh pengguna.

```
FILTER (
  regex(?nama, '". $kata_kunci. "', 'i')
|| regex(?nama_daerahs, '". $kata_kunci. "', 'i')
|| regex(?abstraks, '". $kata_kunci. "', 'i')
|| regex(?kandungans, '". $kata_kunci. "', 'i')
|| regex(?nama_ilmiah, '". $kata_kunci. "', 'i')
|| regex(?divisi, '". $kata_kunci. "', 'i')
|| regex(?subdivisi, '". $kata_kunci. "', 'i')
|| regex(?kelas, '". $kata_kunci. "', 'i')
|| regex(?bangsa, '". $kata_kunci. "', 'i')
|| regex(?suku, '". $kata_kunci. "', 'i')
|| regex(?marga, '". $kata_kunci. "', 'i')
|| regex(?jenis, '". $kata_kunci. "', 'i')
|| regex(?penyakit, '". $kata_kunci. "', 'i')
)
```

Gambar 6. 17 Potongan kode pada method findHerb

Jika berhasil dan ada tanaman yang cocok dengan kata kunci maka akan dikembalikan hasilnya dan kemudian akan ditampilkan hasil dari pencarian tersebut.



Gambar 6. 18 Hasil pencarian dengan kata kunci "wasir"

Pada Gambar 6.18 merupakan penjelasan dari hasil pencarian dengan kata kunci “wasir”, dan terlihat bahwa terdapat hasil pencarian sebanyak 4. Untuk melihat info detail mengenai salah satu hasil pencarian maka pengguna dapat mengklik salah satu dari judul tanaman.

6.3.2. Menu Daftar Tanaman Obat

Menu Daftar Tanaman Obat merupakan salah satu fitur yang harus ada pada aplikasi ensiklopedia, salah satu kriteria ensiklopedia adalah terdapatnya indeks tanaman obat yang diurutkan berdasarkan abjad. Agar lebih memudahkan dalam pencarian tanaman obat. Pada aplikasi tanaman obat ini juga menerapkan hal tersebut, pada Gambar 6.19 di bawah ini merupakan hasilnya.

Daftar Tanaman Obat Menurut Abjad				
A - E:	F - J:	K - O:	P - T:	U - Z:
A	F	K	P	U
B	G	L	Q	V
C	H	M	R	W
D	I	N	S	X
E	J	O	T	Y
				Z

Gambar 6. 19 Indeks Abjad

Kemudian pengguna dapat melihat daftar tanaman obat yang diawali oleh huruf tertentu, sehingga pengguna harus memilih salah satu dari abjad yang ada.

A		
Adem Ati	Akasia	Ambre
Andong	Angsana	Anyelir
Apel	Apokad	Arbei
Aren	Asparagus	Awar-awar

Gambar 6. 20 Daftar Nama Tanaman Obat yang diawali dengan huruf "A"

Pada Gambar 6.20 merupakan salah satu contoh hasil menampilkan daftar tanaman obat yang dimulai dengan huruf "A". Method yang digunakan pada proses menampilkan daftar tanaman obat yang dimulai dengan huruf tertentu akan memanggil method `getHerbByAlphabet($abjad)` pada kelas **Herb**.

```

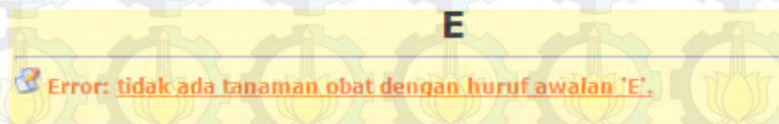
$sparql = "
PREFIX iih: <http://localhost:8080/indonesia_indigenous_herbs/>
SELECT ?nama ?uri WHERE{
  ?uri rdf:type iih:Herb.
  ?uri foaf:name ?nama.
  filter(STRSTARTS(?nama, ".$abjad.")) .
}
ORDER BY ?nama
";

```

Gambar 6. 21 Potongan kode pada method getHerbByAlphabet

Maksud dari query SPARQL di atas adalah mengambil ?nama dan ?uri dimana ?uri merupakan subjek dari yang memiliki predikat `rdf:type foaf:name` dan objek `iih:Herb`, sedangkan ?nama merupakan objek dari ?uri yang berpredikat `foaf:name`. Kemudian dari hasilnya difilter berdasarkan ?nama yang berawalan dengan abjad yang dipilih terakhir diurutkan berdasarkan ?nama.

Namun jika ternyata tidak ada tanaman obat yang diawali dengan huruf abjad, maka akan ada informasi peringatan bahwa tidak ada tanaman obat yang dimulai dengan huruf tersebut. Seperti tampak pada Gambar 6.22 berikut ini.



Gambar 6. 22 Peringatan bahwa tidak ada tanaman obat

Selanjutnya untuk melihat informasi detail mengenai tanaman obat, maka pengguna dapat memilih salah satu dari daftar tanaman obat yang tersedia. Maka akan ditampilkan informasi detail mengenai tanaman obat tersebut.

Akasia

Divisi: Spertmatophyta	
Subdivisi: Angiospermae	
Kelas: Dicotyledonae	
Bangsa: Rosales	
Suku: Leguminosae	
Marga: Acacia	
Jenis: <i>Acacia sieberiana</i> DC.	
Untuk lebih detail klik di sini.	
Nama Daerah:	
Jawa: Akasia (Jawa).	
Kandungan Kimia:	
Akar: Saponin. Bush: Saponin, flavonoida, Polifenol. Daun: Saponin, Flavonoida.	
Keterangan:	
Habitus: Pohon, tinggi 15-20 m. Batang: Tegak, bulat, putih kotor. Daun: Majemuk, berhadapan, menyirip, lonjong, tepi rata, ujung dan pangkal tumpul, panjang 5-20 cm, lebar 1-2 cm, pertulangan menyirip, hijau. Bunga: Majemuk, berkelamin dua, di ketiak daun, kelopak silindris, benang sari silindris, kepala sari bentuk ginjal, mahkota putih, bentuk seperti kuku, putih. Buah: Polong, masih muda hijau setelah tua coklat. Biji: Lonjong, pipih, coklat. Akar: Tunggang, putih kotor.	
Untuk Obat: Demam	
Untuk obat demam dipakai kurang lebih 10 gram akar segar <i>Acacia sieberiana</i> , dicuci, dipotong-potong, direbus dengan 2 gelas air selama 15 menit, dinginkan dan disaring. Hasil saringan diminum dua kali sama banyak pagi dan sore.	

Gambar 6. 23 Informasi detail mengenai Akasia

Pada Gambar 6.23 merupakan informasi detail mengenai tanaman obat Akasia. Pada proses menampilkan halaman ini terdapat beberapa method. Pertama adalah method `getHerbByURI($uri)` pada kelas `Herb`, method tersebut akan memiliki parameter `$uri`, penggunaan parameter `$uri` dikarenakan unik, jadi dimungkinkan untuk menghindari adanya tanaman obat yang memiliki nama yang sama. Selanjutnya untuk menampilkan penyakit yang dapat

disembukan dengan tanaman obat tersebut adalah dengan memanggil method **getDiseaseByHerb(\$herb)** pada kelas Disease dimana method tersebut akan melakukan query dengan parameter **\$herb** yang berarti akan mencari penyakit yang dapat diobati dengan tanaman obat tersebut. Terakhir adalah dalam proses tata cara pengolahan tanaman obat atau resep, method yang dipanggil adalah **getRecipe(\$herb, \$disease)** pada kelas Recipe, jadi method tersebut akan melakukan query untuk mencari resep atau tata cara pengolahan dengan tanaman obat tertentu dan untuk mengobati penyakit tertentu juga, sehingga ada dua parameter, yaitu **\$herb** dan **\$disease**.

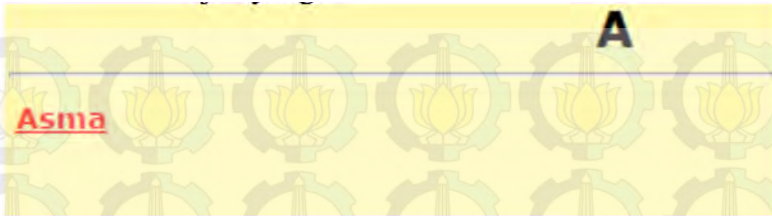
6.3.3. Menu Daftar Penyakit

Menu Daftar Penyakit merupakan salah satu fitur yang harus ada pada aplikasi ensiklopedia, salah satu kriteria ensiklopedia adalah terdapatnya indeks tanaman obat yang diurutkan berdasarkan abjad serta dilengkapi dengan daftar penyakit. Agar lebih memudahkan dalam pencarian tanaman obat. Pada aplikasi tanaman obat ini juga menerapkan hal tersebut, pada Gambar 6.24 di bawah ini merupakan hasilnya.

Daftar Penyakit Menurut Abjad					
A - E:	F - J:	K - O:	P - T:	U - Z:	
A	F	K	P	U	
B	G	L	Q	V	
C	H	M	R	W	
D	I	N	S	X	
E	J	O	T	Y	
				Z	

Gambar 6. 24 Daftar Abjad

Kemudian pengguna dapat melihat daftar penyakit yang diawali oleh huruf tertentu, sehingga pengguna harus memilih salah satu dari abjad yang ada.



Gambar 6. 25 Daftar Penyakit yang diawali dengan huruf "A"

Pada Gambar 6.25 merupakan salah satu contoh hasil menampilkan penyakit yang dimulai dengan huruf "A". Method yang digunakan pada proses menampilkan penyakit yang dimulai dengan huruf tertentu akan memanggil method **getDiseaseByAlphabet(Sabjad)** pada kelas **Disease**. Namun jika ternyata tidak penyakit yang diawali dengan huruf abjad, maka akan ada informasi peringatan bahwa tidak ada penyakit yang dimulai dengan huruf tersebut. Seperti tampak pada Gambar 6.26 berikut ini.



Gambar 6. 26 Peringatan bahawa tidak ada penyakit

Selanjutnya untuk melihat informasi detail mengenai penyakit, maka pengguna dapat memilih salah satu dari daftar penyakit yang tersedia. Maka akan ditampilkan informasi detail mengenai penyakit tersebut.

Batuk

Nama: Batuk

Abstrak:

Batuk bukanlah suatu penyakit. Batuk merupakan mekanisme pertahanan tubuh di saluran pernapasan dan merupakan gejala suatu penyakit atau reaksi tubuh terhadap iritasi di tenggorokan karena adanya lendir, makanan, debu, asap dan sebagainya. Batuk terjadi karena rangsangan tertentu, misalnya debu di reseptor batuk. Kemudian reseptor akan mengalirkan lewat syaraf ke pusat batuk yang berada di otak. Di sini akan memberi sinyal kepada otot-otot tubuh untuk mengeluarkan benda asing tadi, hingga terjadilah batuk.



Untuk lebih detail klik di sini.

Diobati dengan:

Blimbing Legi, Blimbing Wuluh, Druju, Jeruk Keprok, Jeruk Kingkit, Jeruk Nipis, Jeruk Purut, Kapulaga, Kencur, Lobelia, Mengkudu, Palikan Kebo, Randu, Senggugu, Sente, Serpili, Tebu, Tembelele, Temi, Verbaskum, Waron, Waru Gombong

Gambar 6. 27 Informasi detail mengenai Batuk

Pada Gambar 6.27 merupakan informasi detail mengenai penyakit Batuk. Pada proses menampilkan halaman ini terdapat beberapa method. Pertama adalah method **getDiseaseByURI(\$uri)** pada kelas **Disease**, method tersebut akan memiliki parameter **\$uri**, penggunaan parameter **\$uri** dikarenakan unik, jadi dimungkinkan untuk menghindari adanya penyakit yang memiliki nama yang sama. Pada bagian ini merupakan penerapan serta pemanfaatan dari linked data. Pada method **getDiseaseByURI** terdapat query ke sumber data yang berbeda dalam hal ini bersumber dari ontologi yang penulis buat dan ke sumber data lain yaitu DBpedia. Pada Gambar 6.28 di bawah ini merupakan potongan kodenya.

```

Sparql = "
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX iih: <http://localhost:8080/indonesia_indigenous_herbs/>
SELECT ?nama ?abstrak ?url (GROUP_CONCAT(DISTINCT ?nama_tanamans; SEPARATOR=', ') AS ?nama_tanaman)
WHERE {
  <http://localhost:8080/indonesia_indigenous_herbs/".Suri."> foaf:name ?nama.
  ?tanaman iih:treatmentFor <http://localhost:8080/indonesia_indigenous_herbs/".Suri.">.
  ?tanaman foaf:name ?nama_tanamans.
  SERVICE <http://id.dbpedia.org/sparql>
  {
    ?s foaf:isPrimaryTopicOf <http://id.wikipedia.org/wiki/".Suri.">.
    ?s dbpedia-owl:abstract ?abstrak.
    ?s foaf:depiction ?url.
  }
  GROUP BY ?nama ?abstrak ?url
}
"

```

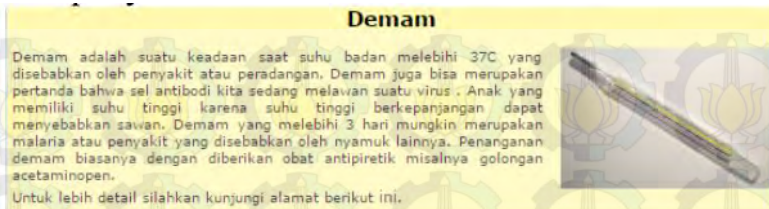
Gambar 6. 28 Potongan kode pada method getDiseaseByURI

Query SPARQL di atas merupakan query terhadap dua sumber yang berbeda yaitu sumber local yang merupakan ontologi tanaman obat yang dibuat, dan sumber lain yaitu id.dbpedia.org. Untuk melakukan query ke sumber lain menggunakan sintaks SERVICE kemudian diikuti dengan alamat dari endpoint-nya.

Dengan pemanfaatan open data ini diharapkan memudahkan dalam pemanfaatan data lebih baik lagi serta tidak perlu membuat data baru terutama data penyakit yang sudah disediakan oleh DBpedia. Serta untuk menampilkan daftar tanaman obat yang dapat digunakan untuk penyakit tersebut.

6.3.4. Menu Informasi Penyakit

Menu ini merupakan fitur dalam menampilkan penjelasan mengenai penyakit, sehingga ada gambaran mengenai penyakit tersebut. Fitur ini juga untuk membuktikan bahwa dalam *Semantic Web* khususnya *linked data* dimungkinkan dapat mengambil data dari banyak sumber data. Pada aplikasi ini penulis mengambil data dari DBpedia untuk informasi mengenai penyakit.



Gambar 6. 29 Contoh informasi mengenai "Demam"

Pada Gambar 6.29 di atas merupakan salah satu contoh informasi mengenai penyakit "Demam". Pada proses ini akan memanggil method `getDiseaseByName($disease)` pada kelas **Disease**.

```
function getDiseaseByName ($disease)
{
    require_once( "lib/sparql.php" );
    $db = sparql_connect( "http://id.dbpedia.org/sparql" );
    if( !$db ) { print sparql_errno() . " : " . sparql_error() . "\n"; exit; }
    sparql_ns( "foaf", "http://xmlns.com/foaf/0.1/" );
    $sparql = "
        select ?nama ?abstrak ?url
        WHERE {
            ?s foaf:isPrimaryTopicOf <http://id.wikipedia.org/wiki/\"$disease.\">.
            ?s rdfs:label ?nama.
            ?s dbpedia-owl:abstract ?abstrak.
            ?s foaf:depiction ?url.
        }";
}
```

Gambar 6. 30 Potongan kode pada method `getDiseaseByName`

Pada Gambar 6.30 menampilkan potongan dari kode pada method `getDiseaseByName`. Jika informasi penyakit tidak tersedia, maka akan muncul pesan peringatan bahwa informasi untuk penyakit tersebut tidak tersedia, seperti tampak pada Gambar 6.31 di bawah ini.



Error: informasi mengenai 'Kulit gatal' masih belum tersedia.

Gambar 6. 31 Pesan error ketika tidak ada informasi penyakit "Kulit gatal"

6.4. Uji Coba

Bagian ini membahas tentang hasil uji coba terhadap aplikasi beserta analisisnya. Adapun uji coba yang dilakukan adalah uji coba Fungsional, dan *Stress Test*.

6.4.1. Fungsional

Uji coba fungsional merupakan uji coba yang dilakukan untuk memastikan fitur-fitur pada aplikasi telah berjalan dengan baik. Uji coba fungsional untuk aplikasi Ensiklopedia Tanaman Obat mengacu pada test case yang telah dibuat pada tahapan desain. Tabel 6.3 berikut adalah hasil uji coba dengan test case yang telah dilakukan. Untuk hasil detail dapat dilihat pada Lampiran F.

Tabel 6. 3 Hasil *Test Case*

Kode	Test Case	Status
TC-01	Melihat Abjad Tanaman Obat	Sukses
TC-02	Melihat Daftar Tanaman Obat	Sukses
TC-03	Melihat Informasi Ilmiah Tanaman Obat	Sukses

TC-04	Melihat Daftar Penyakit Untuk Tiap Tanaman Obat	Sukses
TC-05	Melihat Resep Obat Untuk Tiap Penyakit	Sukses
TC-06	Melihat Abjad Penyakit	Sukses
TC-07	Melihat Daftar Penyakit	Sukses
TC-08	Melihat Abstrak Penyakit	Sukses
TC-09	Melihat Gambar Ilustrasi Penyakit	Sukses
TC-10	Melihat Daftar Tanaman Obat Untuk Tiap Penyakit	Sukses
TC-11	Mencari Tanaman Obat	Sukses

6.4.2. Perbandingan

Pengujian kinerja dari aplikasi ini akan dibandingkan dengan salah satu search engine yang ada yaitu Google. Penulis membandingkan dengan Google karena secara umum orang akan mencari sesuatu yang diinginkan melalui Google, baru kemudian mengunjungi website atau artikel mengenai hal yang dicari tersebut. Skenario yang dilakukan adalah dengan melakukan pencarian. Untuk yang pertama pencarian secara umum dengan kata kunci “jahe”. Pada aplikasi ensiklopedia tanaman obat sendiri setelah dilakukan pencarian menghasilkan hasil seperti pada Gambar 6.32 di bawah ini.



Gambar 6. 32 Hasil pencarian "jahe" pada aplikasi

Sedangkan pencarian pada mesin pencari Google menghasilkan hasil seperti pada Gambar 6.33 di bawah ini.



Gambar 6. 33 Hasil pencarian "jahe" pada Google

Ini menunjukkan bahwa untuk hasil pencarian dengan nama tanaman obat sama-sama valid, yang membedakan adalah

dikarenakan hasil Google lebih banyak karena mengambil dari banyak sumber, sedangkan dalam aplikasi ini untuk tanaman obat dari satu sumber data saja yaitu onotologi tanaman obat.

Selanjutnya dengan mencari tanaman obat dengan kata kunci penyakit maksudnya adalah mencari tanaman obat yang dapat digunakan sebagai obat untuk penyakit tersebut.

Pada aplikasi ensiklopedia tanaman obat untuk mencarinya tinggal dengan menuliskan kata kunci dari nama penyakit, misalkan kata kunci “darah tinggi”. Hasilnya adalah seperti pada Gambar 6.34 berikut.



Gambar 6. 34 Hasil pencarian "darah tinggi" pada aplikasi

Sedangkan pada Google jika kita menuliskan dengan kata kunci “darah tinggi” maka yang muncul adalah informasi mengenai penyakit darah tinggi tersebut, tampak seperti pada Gambar 6.35 di bawah ini



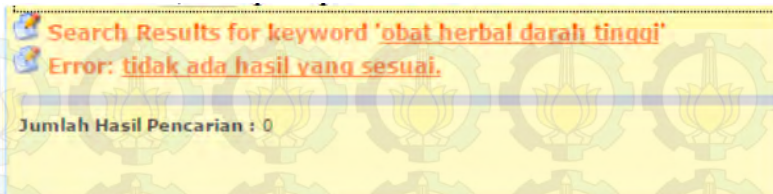
Gambar 6. 35 Hasil pencarian "darah tinggi" pada Google

Sehingga diperlukan kata kunci tambahan “obat herbal darah tinggi”. Hasilnya lebih spesifik dan jelas. Seperti pada Gambar 6.36 di bawah ini.



Gambar 6. 36 Hasil pencarian "obat herbal darah tinggi" pada Google

Namun jika kata kunci yang diterapkan pada Google dilakukan pada aplikasi ensiklopedia tanaman obat maka tidak menemukan hasil, tampak pada Gambar 6.37 di bawah ini.



Gambar 6. 37 Hasil pencarian "obat herbal darah tinggi" pada aplikasi

Perbandingan kinerja dalam hal fitur pencarian dengan fitur pencarian pada Google telah dilakukan.

6.4.3. Verifikasi Endpoint

Pada pengujian terhadap endpoint yang sudah dibangun bertujuan untuk memastikan bahwa data ontologi tanaman obat dapat di query dengan SPARQL. Komputer server dan klien terhubung dalam satu jaringan. Komputer server memiliki alamat IP 10.126.14.114. Selanjutnya melakukan konfigurasi terhadap komputer server dengan menginstall OpenRDF Sesame. Serta mengupload ontologi tanaman obat. Membangun aplikasi sederhana berbasis web dan menggunakan bahasa pemrograman PHP. Pada Gambar 6.38 di bawah ini contoh potongan kode programnya query SPARQL.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX : <http://10.126.14.114:8080/indonesia_indigenous_herbs/>

SELECT DISTINCT ?nama
WHERE {
  ?s rdf:type :Herb.
  ?s foaf:name ?nama.
}
LIMIT 10
```

Gambar 6. 38 Potongan kode query

Berikut hasil dari query yang dijalankan di atas.

nama	nama_ilmiah
Adem Atti	Litsea chinensis Lamk.
Akasia	Acacia sieberiana DC.
Ambre	Geranium radula Cavan.
Andong	Cordyline fruticosa (L.) A. Chev.
Angsana	Pterocarpus indica Willd.

Gambar 6. 39 Hasil untuk nama dan nama ilmiah dari 5 tanaman obat

Pengujian terhadap endpoint dari tanaman obat telah dilakukan dengan melakukan beberapa query.

6.4.4. Stress Testing

Stress testing adalah pengujian/testing untuk mengetahui kemampuan software dalam menangani kondisi yang tidak normal dari sisi volume/kuantitas. Pada uji coba bertujuan untuk mengetahui kemampuan dari endpoint tanaman obat yang telah dibuat dalam menangani permintaan atau request query oleh banyak pengguna dalam waktu yang bersamaan. Dalam melakukan stress testing menggunakan Apache Jmeter sebagai *tool*.

Langkah pertama adalah membuat test plan atau skenario uji coba. Ada tiga skenario secara garis besar. Pertama adalah dengan melihat SPARQL query yang simpel dan tidak kompleks. Kedua adalah dengan SPARQL query yang kompleks atau advance. Ketiga adalah dengan campuran dari query simpel dan kompleks. Tingkat kompleksitas diuji karena dengan semakin kompleks serta banyak query maka secara idelanya akan membutuhkan waktu yang agak lama bagi sistem atau endpoint tersebut dalam menjalankan query tersebut. Tingkat kompleksitas dilihat dari banyaknya query serta penggunaan fungsi-fungsi misalnya fungsi agregasi. Skenario untuk query simpel adalah sistem dalam hal ini harus mampu menjalankan query dari request pengguna sebanyak n dengan waktu yang bersamaan, untuk nilai n awal adalah 10, sehingga diharuskan sistem harus mampu menangani request query dari 10 pengguna sekaligus, untuk membandingkannya maka nilai n akan ditambah untuk mengetahui kemampuan endpoint dalam

menangani request sampai batas maksimalnya. Skenario untuk query advance sama seperti skenario query simpel namun yang membedakan adalah request dari pengguna yaitu query yang agak kompleks. Skenario terakhir adalah gabungan dari simpel dan advance, dimana sistem harus mampu menangani query simpel dan advance dari n pengguna dalam waktu yang bersamaan, nilai n akan diperbesar sampai diperoleh kesimpulan bahwa sistem tidak dapat menangani query dari sebanyak n pengguna tersebut.

Untuk Simple Query berikut langkah-langkahnya:

1. Membuat test plan.

Test Plan

Name: Skenario Stress Test

Comments:

User Defined Variables	
Name	Value

Detail Add Add from Clipboard Remove Up Down

☐ Run Thread Groups consecutively (i.e. run groups one at a time)

☐ Run tearDown Thread Groups after shutdown of main threads

☐ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath Browse... Delete Clear

Gambar 6. 40 Membuat Test Plan

2. Selanjutnya mengatur banyaknya pengguna dalam skenario.

Thread Group

Name: Request Query
Comments:
Action to be taken after a Sampler error:
☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users): 10
 Ramp-Up Period (in seconds): 1
 Loop Count: Forever 1
☐ Delay Thread creation until needed
☐ Scheduler

Gambar 6. 41 Mengatur banyaknya pengguna

3. Selanjutnya mengatur jenis request dan server dari endpoint. Jenis request adalah HTTP.

HTTP Request Defaults

Name: HTTP Request
 Comments:

Web Server
 Server Name or IP: 10.126.1 Port Number: 8080
 Timeouts (milliseconds):
 Connect: Response:

HTTP Request
 Implementation: Protocol (http): Content encoding:
 Path:

Parameters

Send Parameters With the Request:

Name	Value	Encode?	Include Equals?

Detail Add Add from Clipboard Delete Up Down

Gambar 6. 42 Membuat HTTP Request

4. Berikutnya mengatur alamat dari endpoint serta parameter dalam http request, dalam hal ini parameter yang digunakan adalah “query”.

The screenshot shows the 'HTTP Request' tool interface. The 'Name' field is set to 'HTTP Request Simple Query'. The 'Path' field contains the URL: `http://openid-workbench.spoiluresindonesia.id/indigenous_herbs/?query`. The 'Method' is set to 'GET'. The 'Parameters' tab is selected, showing a table with one parameter:

Name	Value	Encoder	Include
query	PREFIX iih: <http://10.126.14.114:8080/indonesia_indigenous_herbs/>		☒

Buttons at the bottom include 'Detail', 'Add', 'Add from Clipboard', 'Delete', 'Up', and 'Down'.

Gambar 6. 43 Mengatur alamat endpoint

5. Parameter “query” yang digunakan adalah seperti pada Gambar 6.44 di bawah ini.

The screenshot shows the 'Detail' view for the 'query' parameter. The 'Name' field is 'query' and the 'Value' field contains the following text:

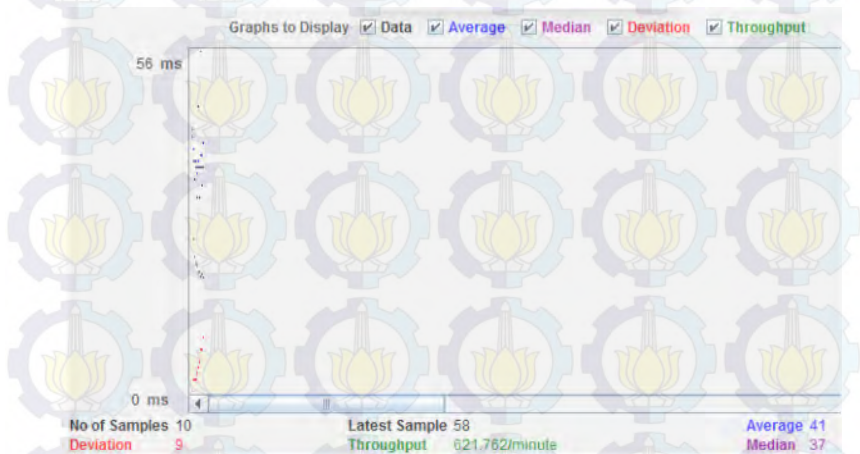
```

1 PREFIX iih: <http://10.126.14.114:8080/indonesia_indigenous_herbs/>
2 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
3
4 SELECT ?nama WHERE {
5     ?s rdf:type iih:Herb.
6     ?s foaf:name ?nama.
7 }

```

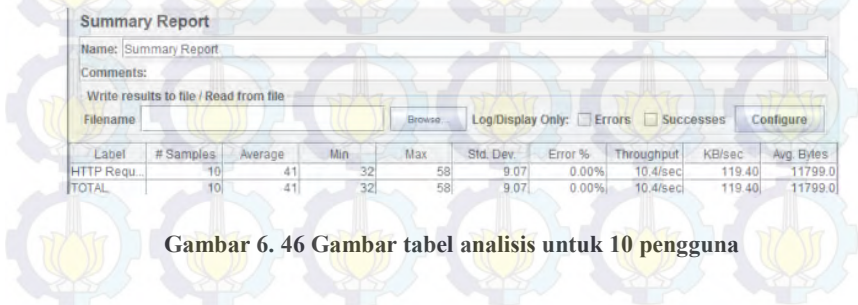
Gambar 6. 44 Parameter simple 'query'

6. Kemudian menambahkan report, pada Gambar 6.45 di bawah ini merupakan grafik dari hasil penanganan request dari 10 pengguna dengan query pada Gambar 6.44 di atas.



Gambar 6. 45 Grafik untuk 10 pengguna

7. Gambar 6.46 di bawah ini ringkasan dari grafik pada Gambar 6.45 di atas.



Gambar 6. 46 Gambar tabel analisis untuk 10 pengguna

8. Pada Gambar 6.47 dan Gambar 6.48 di bawah ini merupakan hasil report untuk 50 pengguna yang melakukan request query.



Gambar 6. 47 Grafik untuk 50 pengguna

Summary Report

Name: Summary Report

Comments:

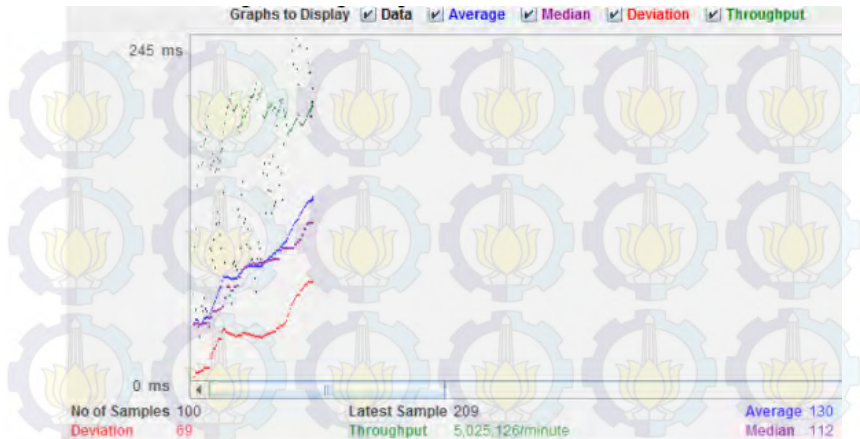
Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Reqs.	50	69	23	205	48.04	0.00%	49.4/sec	568.73	11799.0
TOTAL	50	69	23	205	48.04	0.00%	49.4/sec	568.73	11799.0

Gambar 6. 48 Gambar tabel analisis untuk 50 pengguna

9. Pada Gambar 6.49 dan Gambar 6.50 di bawah ini merupakan hasil report untuk 100 pengguna yang



Gambar 6. 49 Grafik untuk 100 pengguna

Summary Report

Name: Summary Report

Comments:

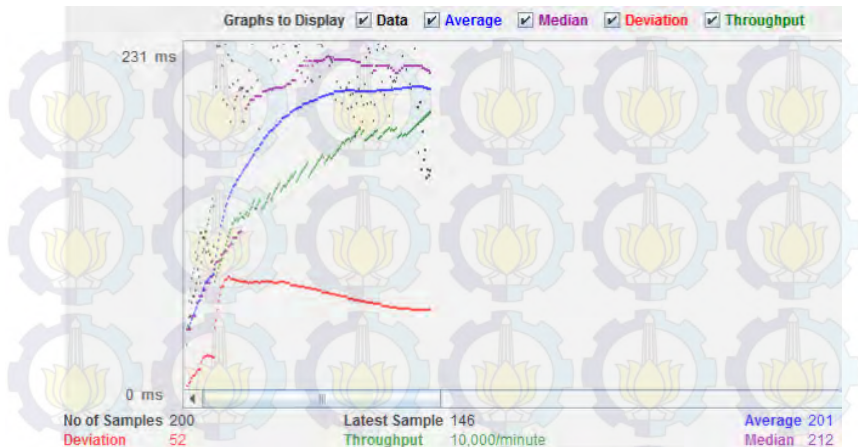
Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ.	100	130	30	271	69.41	0.00%	83.8/sec	965.03	11799.0
TOTAL	100	130	30	271	69.41	0.00%	83.8/sec	965.03	11799.0

Gambar 6. 50 Gambar tabel analisis untuk 100 pengguna

10. Pada Gambar 6.51 dan Gambar 6.53 di bawah ini merupakan hasil report untuk 200 pengguna yang melakukan request query.



Gambar 6. 51 Grafik untuk 200 pengguna

Summary Report

Name: Summary Report

Comments:

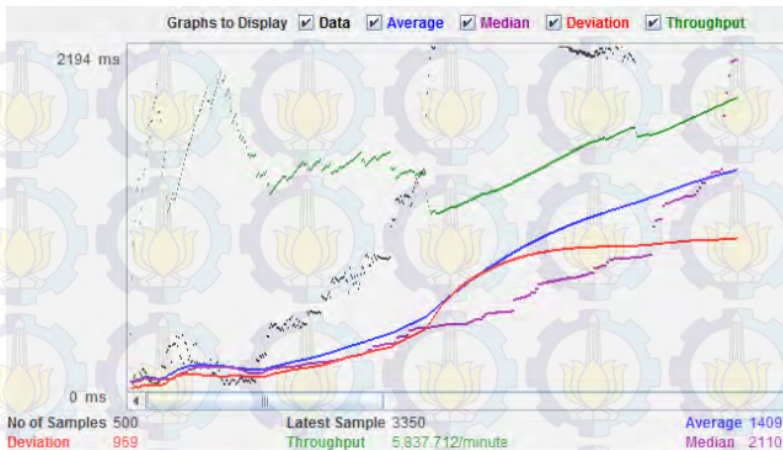
Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Reqs.	200	201	39	304	52.43	0.00%	166.7/sec	1920.41	11799.0
TOTAL	200	201	39	304	52.43	0.00%	166.7/sec	1920.41	11799.0

Gambar 6. 52 Gambar tabel analisis untuk 200 pengguna

11. Pada Gambar 6.53 dan Gambar 6.54 di bawah ini merupakan hasil report untuk 500 pengguna yang melakukan request query.



Gambar 6. 53 Grafik untuk 500 pengguna

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ...	500	1409	35	3358	969.31	0.00%	97.3/sec	1121.69	11805.4
TOTAL	500	1409	35	3358	969.31	0.00%	97.3/sec	1121.69	11805.4

Gambar 6. 54 Gambar tabel analisis untuk 500 pengguna

12. Ketika user ditambah menjadi 1000 waktu response sangat lama dan aplikasi Jmeter mengalami not responding.

Untuk Advance Query berikut langkah-langkahnya:

1. Parameter “query” yang digunakan adalah seperti pada Gambar 6.55 di bawah ini.

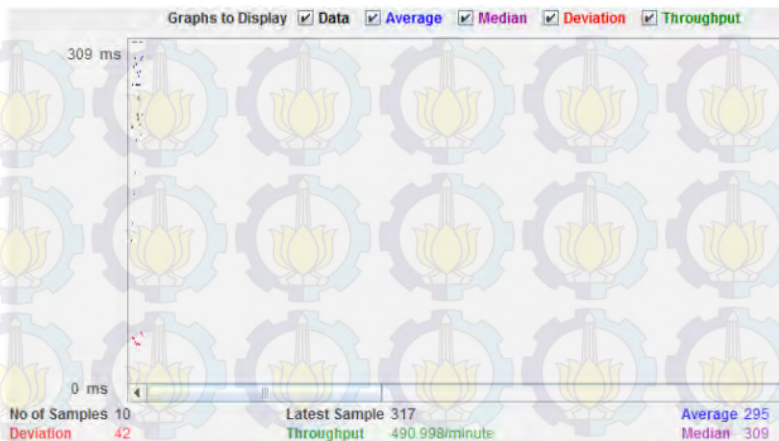
```

Name: query
Value
1 PREFIX iih: <http://10.126.14.114:8080/indonesia_indigenous_herbs/>
2 PREFIX osspr: <http://data.ordnancesurvey.co.uk/ontology/spatialrelations/>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX biol: <http://purl.org/NET/biol/ns#>
5 PREFIX botany: <http://ontology.es/biol/botany#>
6 PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
7 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
8
9 SELECT ?nama ?nama_ilmiah ?divisi ?subdivisi ?kelas ?bangsa ?suku ?marga ?jenis
10 ?gambar (GROUP_CONCAT(DISTINCT ?abstraks; SEPARATOR=' ') AS ?abstrak) (
11 GROUP_CONCAT(DISTINCT ?nama_daerahs; SEPARATOR=' ') AS ?nama_daerah) (
12 GROUP_CONCAT(DISTINCT ?kandungans; SEPARATOR=' ') AS ?kandungan) WHERE{
13
14   ?s rdf:type iih:Herb.
15   ?s foaf:name ?nama.
16   ?s foaf:givenName ?nama_daerahs.
17   ?s biol:name ?nama_ilmiah.
18   ?s botany:division ?divisi.
19   ?s botany:subdivision ?subdivisi.
20   ?s botany:class ?kelas.
21   ?s botany:order ?bangsa.
22   ?s botany:family ?suku.
23   ?s botany:genus ?marga.
24   ?s botany:species ?jenis.
25   ?s foaf:depiction ?gambar.
26   ?s dbpedia-owl:abstract ?abstraks.
27   ?s osspr:contains ?kandungans.
28 }
29
30 GROUP BY ?nama ?nama_ilmiah ?divisi ?subdivisi ?kelas ?bangsa ?suku ?marga ?
31 jenis ?gambar

```

Gambar 6. 55 Parameter advance 'query'

2. Kemudian menambahkan report, pada Gambar 6.56 di bawah ini merupakan grafik dari hasil penanganan request dari 10 pengguna dengan query pada Gambar 6.55 di atas.



Gambar 6. 56 Grafik untuk 10 pengguna

3. Gambar 6.57 di bawah ini ringkasan dari grafik pada Gambar 6.56 di atas.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

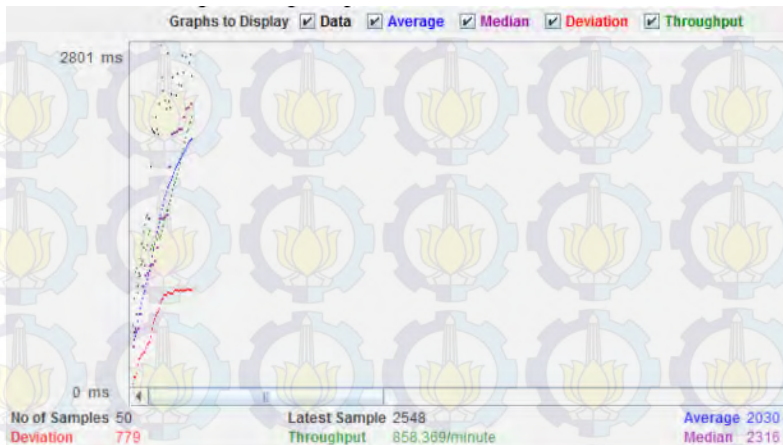
Filename: Browse...

Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ...	10	295	231	381	42.73	0.00%	8.2/sec	1580.01	197711.0
TOTAL	10	295	231	381	42.73	0.00%	8.2/sec	1580.01	197711.0

Gambar 6. 57 Gambar tabel analisis untuk 10 pengguna

4. Pada Gambar 6.58 dan Gambar 6.59 di bawah ini merupakan hasil report untuk 50 pengguna yang melakukan request query.



Gambar 6. 58 Grafik untuk 50 pengguna

Summary Report

Name: Summary Report

Comments:

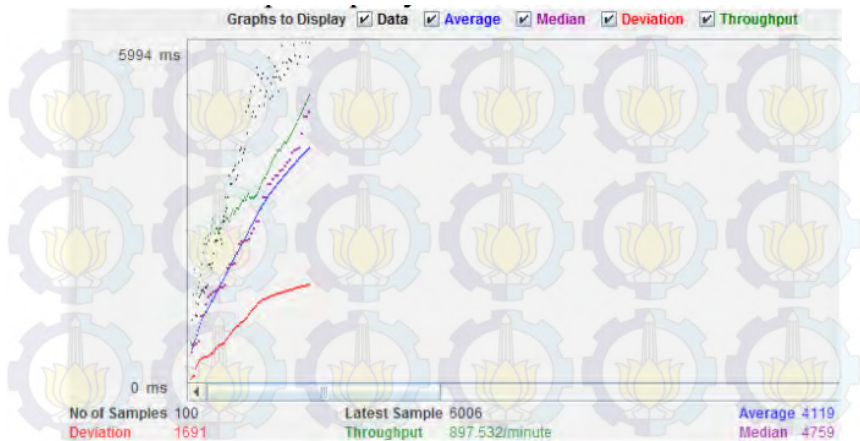
Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ.	50	2030	314	3020	779.83	0.00%	14.3/sec	2762.19	197711.0
TOTAL	50	2030	314	3020	779.83	0.00%	14.3/sec	2762.19	197711.0

Gambar 6. 59 Gambar tabel analisis untuk 50 pengguna

5. Pada Gambar 6.60 dan Gambar 6.61 di bawah ini merupakan hasil report untuk 100 pengguna yang



Gambar 6. 60 Grafik untuk 100 pengguna

Summary Report

Name: Summary Report

Comments:

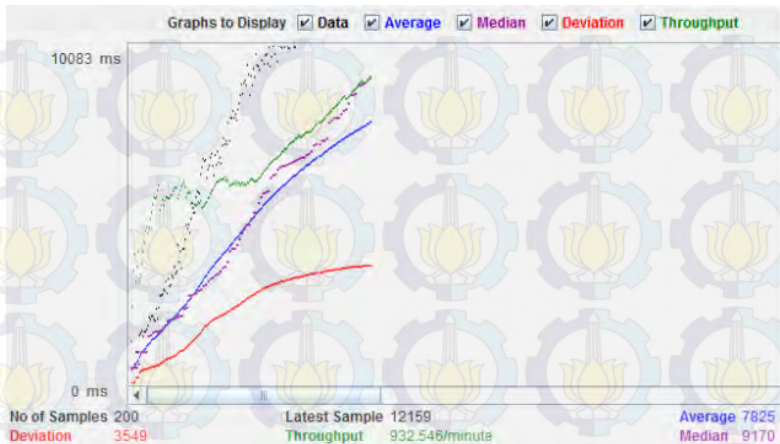
Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ.	100	4119	507	6325	1691.38	0.00%	15.0/sec	2888.21	197711.0
TOTAL	100	4119	507	6325	1691.38	0.00%	15.0/sec	2888.21	197711.0

Gambar 6. 61 Gambar tabel analisis untuk 100 pengguna

6. Pada Gambar 6.62 dan Gambar 6.63 di bawah ini merupakan hasil report untuk 200 pengguna yang melakukan request query.



Gambar 6. 62 Grafik untuk 50 pengguna

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

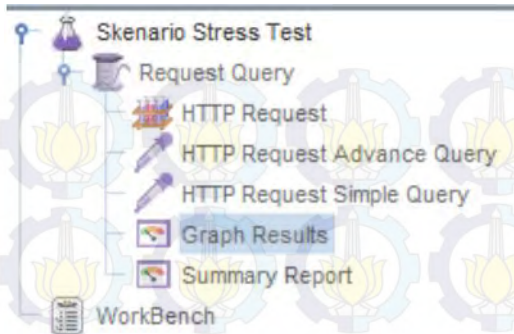
Label	# Samples	Average	Min	Max	Std Dev	Error %	Throughput	KB/sec	Avg Bytes
HTTP Requ...	200	7825	457	12630	3549.58	0.00%	15.5/sec	3000.98	197716.9
TOTAL	200	7825	457	12630	3549.58	0.00%	15.5/sec	3000.98	197716.9

Gambar 6. 63 Gambar tabel analisis untuk 200 pengguna

7. Ketika user ditambah menjadi 500 waktu response sangat lama dan aplikasi Jmeter mengalami not responding.

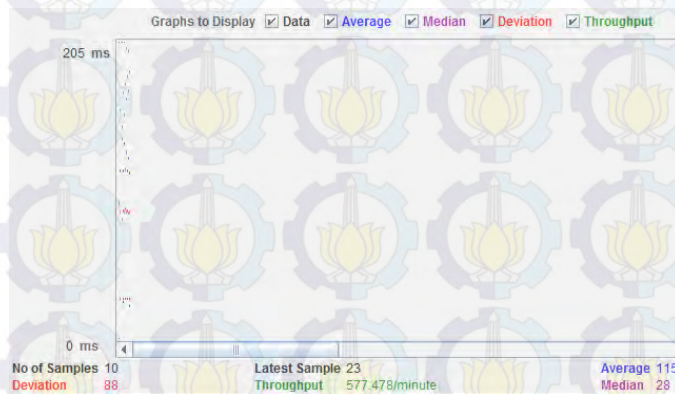
Untuk query simpel dan kompleks berikut langkah-langkahnya:

1. Pada pengaturan HTTP request dibuat sebanyak 2 tipe request yaitu simple dan advance tampak seperti pada Gambar 6.64 di bawah ini.



Gambar 6. 64 Membuat skenario

2. Kemudian menambahkan report, pada Gambar 6.65 di bawah ini merupakan grafik dari hasil penanganan request dari 10 pengguna dengan query pada Gambar 6.64 di atas.



Gambar 6. 65 Grafik untuk 10 pengguna

3. Gambar 6.66 di bawah ini ringkasan dari grafik pada Gambar 6.65 di atas.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

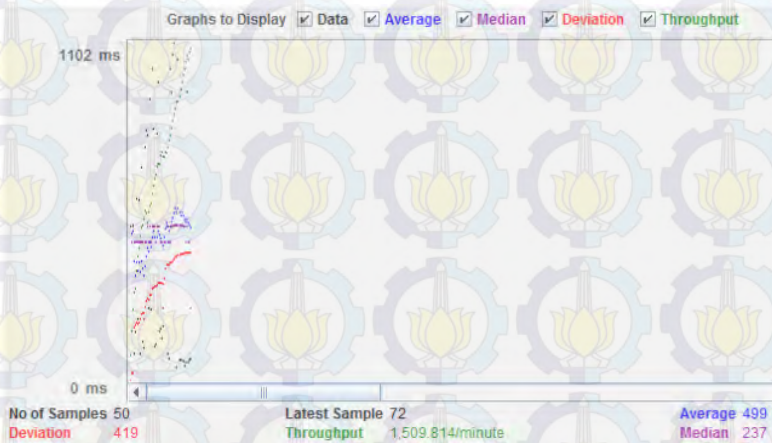
Filename Browse...

Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ...	5	204	199	211	4.18	0.00%	5.0/sec	955.83	197711.0
HTTP Requ...	5	26	23	28	1.96	0.00%	6.1/sec	69.92	11799.0
TOTAL	10	115	23	211	88.96	0.00%	9.6/sec	984.60	104755.0

Gambar 6. 66 Gambar tabel analisis untuk 10 pengguna

4. Pada Gambar 6.67 dan Gambar 6.68 di bawah ini merupakan hasil report untuk 50 pengguna yang melakukan request query.



Gambar 6. 67 Grafik untuk 50 pengguna

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

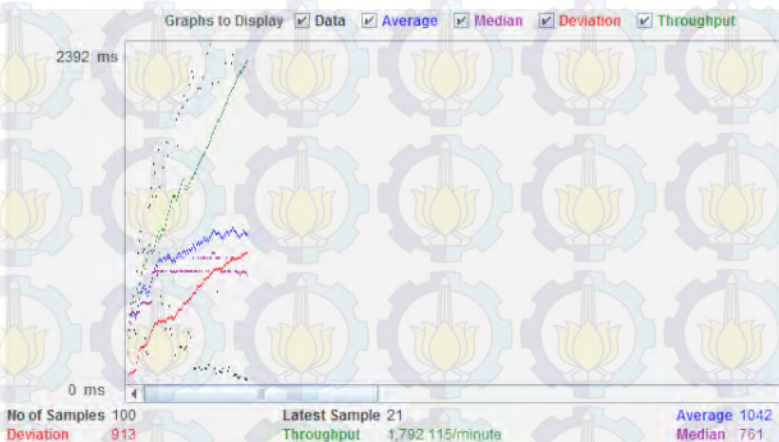
Filename: Browse...

Log/Display Only: ☐ Errors ☒ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Requ...	25	885	454	1319	226.36	0.00%	13.1/sec	2529.84	197711.0
HTTP Requ...	25	113	45	237	56.03	0.00%	16.9/sec	195.30	11799.0
TOTAL	50	499	45	1319	419.62	0.00%	25.2/sec	2574.23	104755.0

Gambar 6. 68 Gambar tabel analisis untuk 50 pengguna

5. Pada Gambar 6.69 dan Gambar 6.70 di bawah ini merupakan hasil report untuk 100 pengguna yang melakukan request query.



Gambar 6. 69 Grafik untuk 100 pengguna

Summary Report									
Name: Summary Report									
Comments:									
Write results to file / Read from file									
Filename						Log/Display Only:	☐ Errors	☐ Successes	Configure
Label	# Samples	Average	Min	Max	Std. Dev	Error %	Throughput	KB/sec	Avg Bytes
HTTP Requ...	100	9993	619	12148	2300.33	0.00%	8.1/sec	1562.24	197711.0
HTTP Requ...	100	515	59	3625	673.77	0.00%	8.5/sec	97.47	11799.0
TOTAL	200	5254	59	12148	6032.61	0.00%	16.1/sec	1643.50	104755.0

Gambar 6. 72 Gambar tabel analisis untuk 200 pengguna

7. Ketika user ditambah menjadi 500 waktu response sangat lama dan aplikasi Jmeter tidak merespon.

6.5. Pemanfaatan Ontologi

Bagian ini menjelaskan mengenai pemanfaatan dari ontology tanaman obat yang telah dibuat. Jadi bagian ini memberitahukan kepada pihak lain yang ingin memanfaatkan data tanaman obat pada ontology tanaman obat. Ontologi tanaman obat ini masih belum dipublikasikan secara online. Namun sudah dipublikasikan secara local. Dianggap ontology tanaman obat ini sudah dipublikasikan secara online dan memiliki alamat

http://10.126.14.114:8080/indonesia_indigenous_herbs/

seperti yang sudah dijelaskan pada sub bab 6.4.3 Verifikasi Endpoint. Kemudian berikut beberapa persyaratan agar dapat melakukan query ke endpoint tanaman obat.

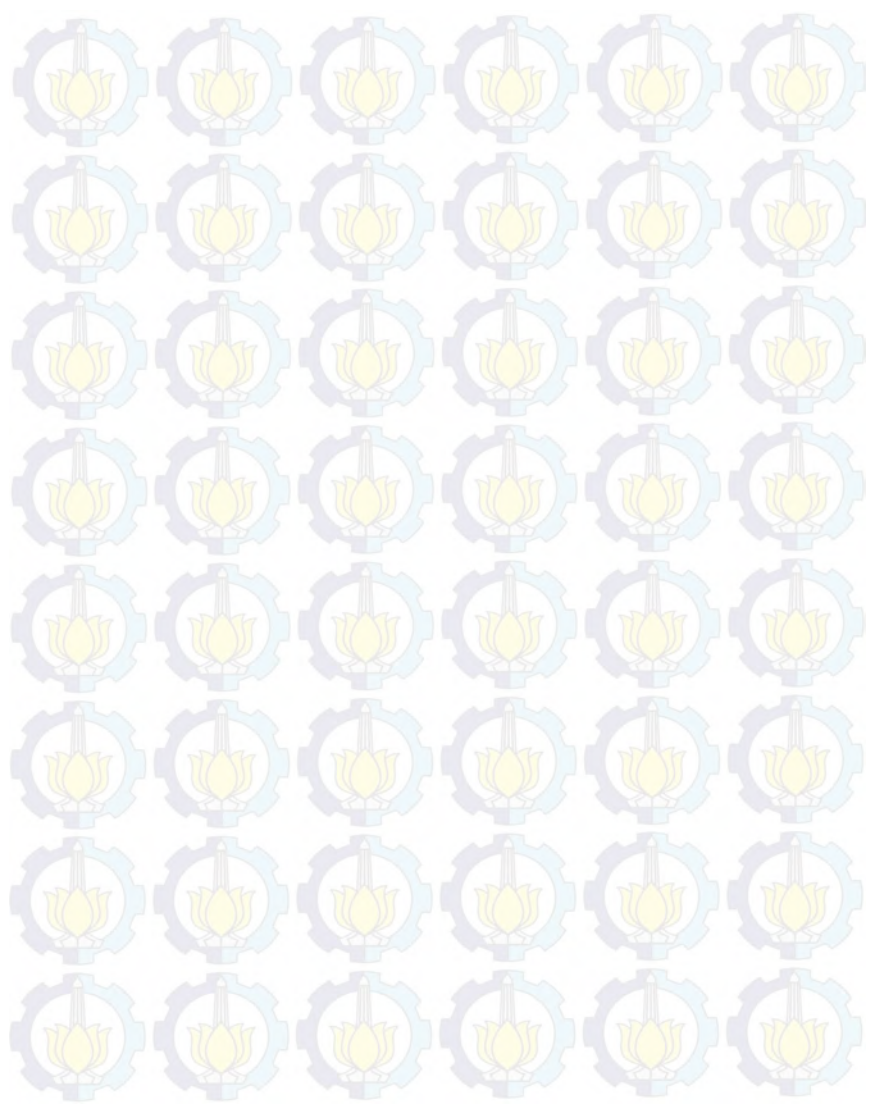
1. Memahami struktur query SPARQL. SPARQL sudah menjadi query dalam linked data yang telah disepakati bersama, jadi sebelum memulai melakukan query harus memahami terlebih dahulu SPARQL tersebut.
2. Memilih bahasa pemrograman aplikasi dan menyesuaikan dengan library yang akan dipakai nantinya dalam menangani pertukaran data dalam query SPARQL. Misalnya untuk PHP dikenal beberapa

library seperti Sparql PHP-Lib, ARC. Untuk Phyton misalnya SPARQL-Wrapper.

3. Melakukan konfigurasi endpoint pada library yang dipakai dan ditujukan pada alamat endpoint tanaman obat yang telah disebutkan di atas.
4. Menuliskan query SPARQL. Disediakan interface yang menggambarkan property-property yang ada pada ontology tanaman obat dengan alamat http://10.126.14.114:8080/indonesia_indigenous_herbs/. Jadi sebelum memulai query sebaiknya melihat terlebih dahulu struktur dalam ontology tanaman obat agar tidak terdapat kesalahan dalam melakukan query.
5. Output dari hasil query dapat berupa json dan xml, jadi dapat dipilih terlebih dahulu, namun secara default outputnya berupa json. Sampai langkah ini data dalam format json tersebut dapat digunakan sesuai dengan kebutuhan.

Langkah-langkah di atas diperuntukkan bagi pengguna yang akan memanfaatkan data dari ontology tanaman obat. Pengguna juga dapat menggabungkan beberapa sumber data atau endpoint, seperti dengan id.dbpedia.org. Pada pembuatan aplikasi ensiklopedia ini penulis juga menggabungkan dengan id.dbpedia.org. Jadi dalam satu query SPARQL dimungkinkan ditujukan kepada banyak dataset atau endpoint yang tersedia.

(halaman ini sengaja dikosongkan)



BAB VII

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dari uji coba serta analisa dari hasil uji coba yang telah dilakukan.

7.1. Fungsional

Pada uji fungsional terdapat 11 test case yang harus terpenuhi dapat dilihat pada Lampiran F.

Pada Test Case 02 Melihat Daftar Tanaman Obat terdapat scenario yang dilakukan, ketika abjad A yang dipilih maka akan muncul daftar tanaman obat yang berawalan khuruf A, namun jika khuruf yang dipilih tidak ada tanaman obat yang sesuai maka akan ditampilkan bahwa tanaman obat dengan khuruf abjad yang dipilih tersebut(abjad E) tidak ada. Test Case ini terpenuhi.

Pada Test Case 03 Informasi Ilmiah Tanaman Obat, Test Case 04 Melihat Daftar Penyakit Untuk Tiap Tanaman Obat, Test Case 05 Melihat Resep Obat Untuk Tiap Penyakit diharuskan ketika salah satu tanaman obat dipilih maka akan ditampilkan informasi detail mengenai tanaman obat tersebut, namun jika tidak ada informasi detail maka akan ditampilkan informasi peringatan bahwa informasi mengenai tanaman obat tersebut masih belum tersedia. Pada Test Case ini dipilih “Akasia”, dan ditampilkan informasi mengenai “Akasia”. Test Case ini terpenuhi.

Pada Test Case 07 Melihat Daftar Penyakit terdapat skenario yang dilakukan, ketika abjad A yang dipilih maka akan muncul daftar penyakit yang berawalan khuruf A, namun jika khuruf yang dipilih tidak ada penyakit yang sesuai maka akan ditampilkan bahwa penyakit dengan khuruf abjad yang dipilih tersebut(abjad F) tidak ada. Test Case ini terpenuhi.

Pada Test Case 08 Melihat Abstrak Penyakit, Test Case 09 Melihat Gambar Ilustrasi Penyakit, Test Case 10 Melihat Daftar Tanaman Obat Untuk Tiap Penyakit diharuskan ketika salah satu penyakit dipilih maka akan ditampilkan informasi detail mengenai penyakit tersebut, namun jika tidak ada informasi detail maka akan ditampilkan informasi peringatan bahwa informasi

mengenai penyakit tersebut masih belum tersedia. Pada Test Case ini dipilih “Demam”, dan ditampilkan informasi mengenai “Demam”, ketika “Kulit Gatal” dipilih ditampilkan pesan peringatan bahwa informasi mengenai penyakit “Kulit Gatal” masih belum tersedia. Test Case ini terpenuhi.

Pada Test Case 11 Mencari Tanaman Obat, diharuskan ketika pengguna memasukkan kata kunci maka akan ditampilkan daftar tanaman obat yang sesuai dengan kata kunci tersebut, baik informasi mengenai namanya, kandungannya, serta penyakit yang dapat diobati dengan tanaman obat tersebut. Pada scenario yang dilakukan ketika kata kunci “wasir” dimasukkan maka muncul beberapa tanaman obat yang dapat digunakan untuk penyakit “wasir” tersebut sebanyak 4 tanaman obat. Ketika menggunakan kata kunci “adasdaffa” ditampilkan pesan bahwa tidak ada yang cocok dengan kata kunci tersebut. Serta ketika kata kunci belum diisi muncul pesan peringatan bahwa kata kunci belum diisi dan harus diisi terlebih dahulu. Test Case ini terpenuhi.

Berdasarkan hasil uji fungsional pada Tabel 6.3, semua test case sukses dan terpenuhi. Untuk informasi detail mengenai test case dapat dilihat pada Lampiran F. Oleh karena itu dapat disimpulkan bahwa Aplikasi Ensiklopedia Tanaman Obat secara fungsional telah berjalan dengan baik.

7.2. Perbandingan

Percobaan pada perbandingan pencarian didapatkan hasil yang sama namun pada pencarian Google diperlukan kata kunci yang lebih spesifik. Sedangkan pada aplikasi ensiklopedia tanaman obat harus dengan kata kunci sesuai domainnya, misalkan nama tanaman obat, nama penyakit. Pada Tabel 7.1 berikut perbandingan kinerja antara aplikasi ensiklopedia tanaman obat dengan Google.

Tabel 7. 1 Perbandingan aplikasi ensiklopedia tanaman obat dengan

No	Aplikasi Ensiklopedia Tanaman Obat	Google
1	Kata kunci dalam melakukan pencarian hanya yang berhubungan dengan domain, misalkan naman tanaman obat, nama penyakit dan lain sebagainya	Kata kunci dalam melakukan pencarian diharuskan dengan penambahan kata-kata sehingga hasil pencarian lebih spesifik.
2	Fitur pencarian sederhana hanya dengan mencocokkan kata kunci dengan data yang ada.	Adanya beberapa fitur pencarian agar yang dicari bisa lebih spesifik, misalkan diurutkan dengan tanggal, filer bahasa dan lain sebagainya.
3	Algoritma pencarian sederhana ikarenakan memanfaatkan fitur REGEX untuk mencocokkan dengan data.	Algoritma pencarian sudah lebih baik, karena dalam pencarian dengan kata kunci, fungsi OR AND, dan lain sebagainya.

4	Fokus pada domain tanaman obat sehingga, hasil pencarian dapat dipastikan berhubungan	Hasilnya pencarian banyak dikarenakan Google mengambil dari banyak sumber sehingga, apa yang dicari kadang tidak ada hubungannya
---	---	--

Perbandingan telah dilakukan dan dapat disimpulkan bahwa Google memiliki fitur yang lebih baik dalam hal pencarian, namun aplikasi lebih unggul dalam data karena domain yang dipakai fokus pada tanaman obat sehingga apa yang dicari pasti ada hubungannya dengan tanaman obat, namun Pada Google masih ada beberapa hasil pencarian yang kurang sesuai dengan yang dicari.

7.3. Verifikasi Endpoint

Pada pengecekan endpoint yang dijelaskan pada Bab 6.4.3 didapatkan hasil yang sesuai dengan query, sehingga dapat disimpulkan bahwa endpoint berjalan dengan baik dan dapat dilakukan query oleh pengguna lain.

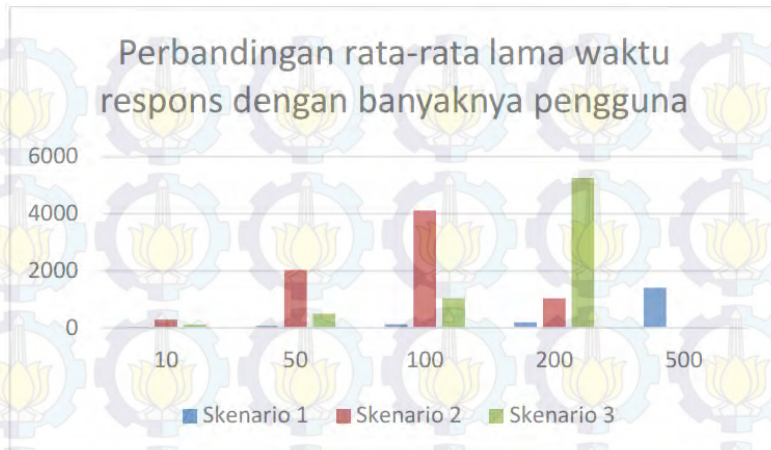
7.4. Stress Testing

Berikut perbandingan antara ketiga skenario dalam stress testing. Dapat dilihat pada Tabel 7.2 di bawah ini.

Tabel 7. 2 Perbandingan jumlah pengguna dengan rata-rata lamanya waktu response

Skenario	Jumlah Pengguna				
	10	50	100	200	500
1	41 ms	69 ms	130 ms	201 ms	1409 ms
2	295 ms	2030 ms	4119 ms	7825 ms	-
3	115 ms	499 ms	1042 ms	5254 ms	-

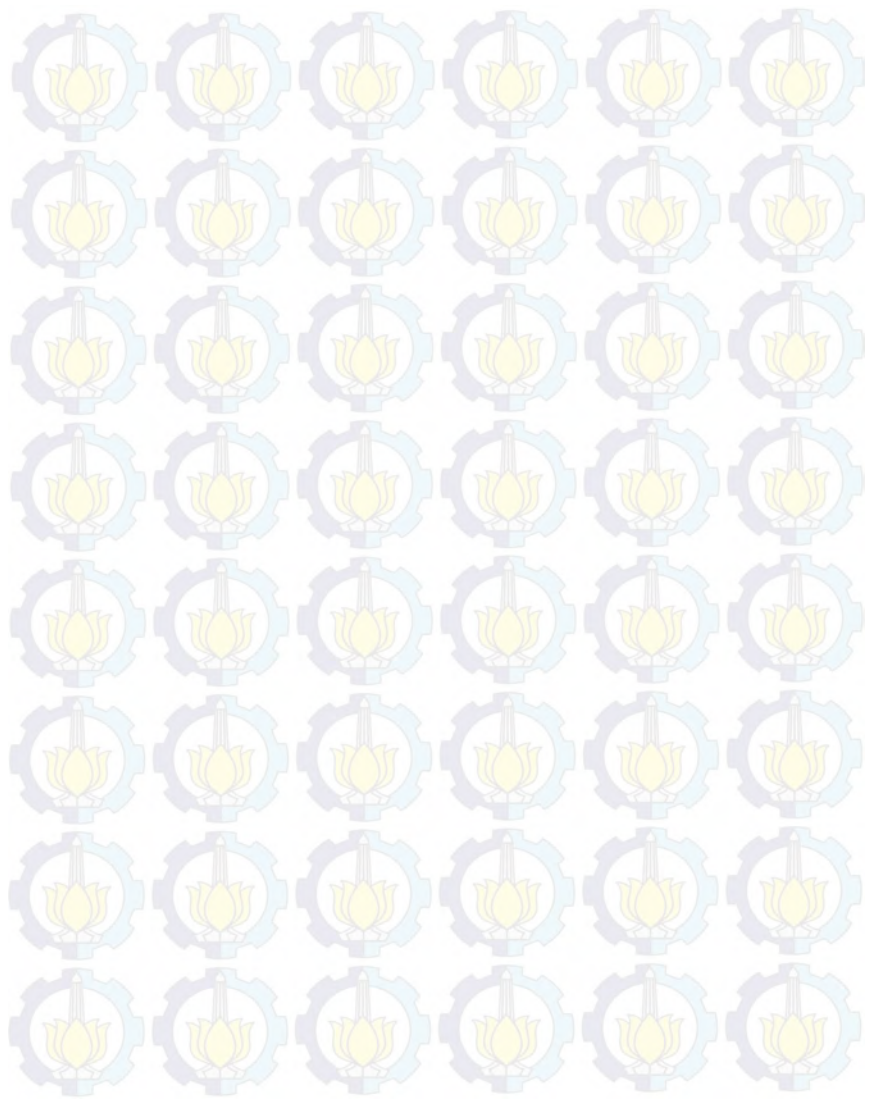
Dari tabel 7.2 di atas dapat direpresentasikan dengan grafik di bawah ini.



Gambar 7. 1 Grafik perbandingan hasil stres test

Berdasarkan hasil uji stress testing yang telah dilakukan, endpoint tanaman obat sanggup menyelesaikan tiga skenario yang dibuat. Endpoint dapat menangani request dari pengguna berupa query SPARQL, walaupun semakin banyak pengguna yang melakukan query dan query yang digunakan semakin kompleks membuat waktu tunggu semakin lama. Ini terjadi dikarenakan lamanya endpoint dalam menangani query yang dilakukan, tentu kejadian ini normal. Dari analisa dapat diprediksi server endpoint mengalami penurunan dalam kemampuan menangani request yang semakin banyak. Dapat disimpulkan bahwa endpoint tanaman obat secara performa sudah cukup baik dan siap digunakan oleh pengguna lain.

(halaman ini sengaja dikosongkan)



BAB VIII

KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan mengenai kesimpulan yang didapat dari hasil pengerjaan Tugas Akhir ini serta saran untuk penelitian pada Tugas Akhir sehingga diharapkan penelitian selanjutnya yang berhubungan dengan permasalahan yang di bahas pada Tugas Akhir ini menjadi lebih baik lagi.

8.1. Kesimpulan

Berdasarkan hasil pengerjaan tugas akhir ini, maka dapat disimpulkan sebagai berikut:

1. Aplikasi Ensiklopedia Tanaman Obat berbasis web menggunakan ontologi sebagai basis datanya berhasil dibuat. Aplikasi Ensiklopedia Tanaman obat memungkinkan pengguna untuk mengenali tanaman obat yang ada di Indonesia serta manfaat tanaman obat tersebut. Pengguna juga dapat melakukan pencarian tanaman obat melalui indeks. Berdasarkan hasil uji coba fungsional, semua kebutuhan fungsional aplikasi sudah terpenuhi.
2. Ontologi Tanaman Obat Indonesia telah dibuat dengan metode Methontology melalui beberapa tahapan dalam pembuatan ontologi. Hasil uji coba ontologi tersebut menunjukkan bahwa pengguna dapat mengakses dan mengambil data ontologi. Serta dari hasil pengujian ketahanan atau *stress testing* menunjukkan bahwa *endpoint* tanaman obat yang dibuat mampu menangani permintaan dari banyak pengguna dalam waktu yang bersamaan. Sehingga ontologi tanaman obat ini siap dan dapat digunakan oleh pengguna yang membutuhkan data tanaman obat.
3. Representasi data tanaman obat dalam bentuk OWL dapat menjadi alternatif penyajian pengetahuan tentang tanaman obat, sehingga memudahkan pengguna dalam

melakukan pencarian informasi tanaman obat dan penyakit.

8.2. Saran

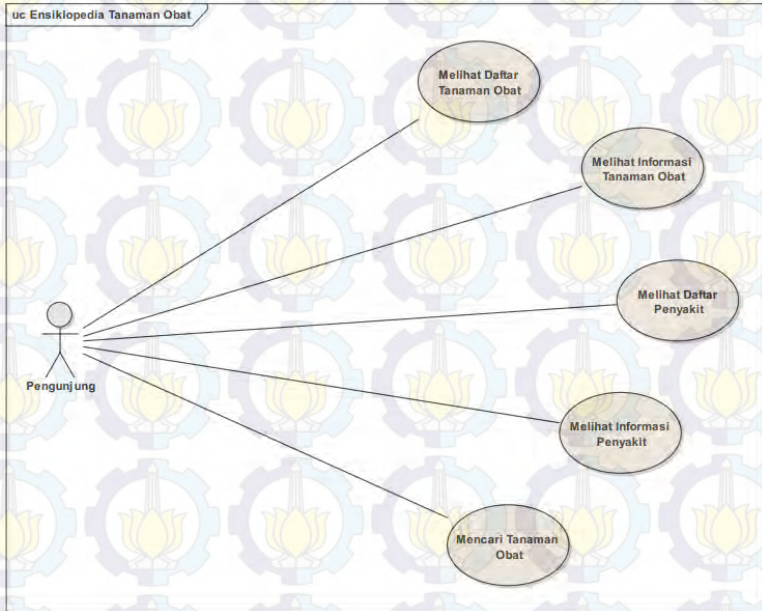
Aplikasi Ensiklopedia Tanaman Obat dan Ontologi Tanaman Obat Indonesia masih masih terdapat beberapa hal yang perlu diperbaiki dan dikembangkan, maka terdapat beberapa saran yang dapat dipertimbangkan untuk penelitian lebih lanjut yaitu sebagai berikut:

1. Fitur pada aplikasi ensiklopedia tanaman obat dapat ditambah lagi. Fitur penambahan data tanaman obat diperlukan agar pengguna dapat menambahkan informasi tanaman obat yang masih belum ada. Kemudian sistem melakukan penambahan data terhadap ontologi tanaman obat dari data yang ditambahkan oleh pengguna, sehingga ontologi tanaman obat semakin kaya dan lengkap.
2. Informasi mengenai penyakit dapat diperkaya dengan menghubungkannya dengan ontologi penyakit yang sudah ada selain DBpedia.

LAMPIRAN-LAMPIRAN

Lampiran A

Use Case Diagram



Gambar A. 1 Use Case Diagram

Use Case Skenario

Tabel A. 1 Skenario Melihat Daftar Tanaman Obat

Use Case Code	UC-01
Use Case Name	Melihat Daftar Tanaman Obat
Use Case User	Pengunjung
Scenarios	
Basic	Pengunjung memilih menu Daftar Tanaman Obat. Sistem menampilkan daftar abjad dari A sampai Z. Kemudian pengunjung memilih salah satu abjad. Sistem mencocokkan abjad yang dipilih oleh pengunjung, kemudian sistem menampilkan daftar tanaman obat yang berawalan abjad dengan abjad yang dipilih
Daftar Tanaman Obat tidak ada	Sistem menampilkan pesan bahwa tanaman obat tidak ada.

Tabel A. 2 Skenario Melihat Informasi Tanaman Obat

Use Case Code	UC-02
Use Case Name	Melihat Informasi Tanaman Obat
Use Case User	Pengunjung
Scenarios	
Basic	Pengunjung memilih tanaman obat untuk melihat informasi detail. Kemudian sistem menampilkan informasi detail mengenai tanaman

	obat.
Tidak ada detail tanaman obat	Sistem menampilkan pesan bahwa tidak ada informasi detail mengenai tanaman obat yang dipilih.

Tabel A. 3 Skenario Melihat Daftar Penyakit

Use Case Code	UC-03
Use Case Name	Melihat Daftar Penyakit
Use Case User	Pengunjung
Scenarios	
Basic	Pengunjung memilih menu Daftar Tanaman Penyakit. Sistem menampilkan daftar abjad dari A sampai Z. Kemudian pengunjung memilih salah satu abjad. Sistem mencocokkan abjad yang dipilih oleh pengunjung, kemudian sistem menampilkan daftar penyakit yang berawalan abjad dengan abjad yang dipilih.
Daftar Penyakit tidak ada	Sistem menampilkan pesan bahwa penyakit tidak ada.

Tabel A. 4 Skenario Melihat Informasi Penyakit

Use Case Code	UC-04
Use Case Name	Melihat Informasi Penyakit

Use Case User	Pengunjung
Scenarios	
Basic	Pengunjung memilih tanaman obat untuk melihat informasi detail. Kemudian sistem menampilkan informasi mengenai penyakit. Pengunjung memilih penyakit. Selanjutnya sistem menampilkan informasi detail mengenai penyakit. Pengunjung memilih penyakit untuk melihat informasi detail. Kemudian sistem menampilkan informasi detail mengenai penyakit.
Informasi Penyakit tidak ada	Sistem menampilkan pesan bahwa informasi detail penyakit tidak ada.

Tabel A. 5 Skenario Mencari Tanaman Obat

Use Case Code	UC-05
Use Case Name	Mencari Tanaman Obat
Use Case User	Pengunjung
Scenarios	
Basic	Pengunjung membuka halaman utama dari aplikasi ensiklopedia tanaman obat. Sistem menampilkan halaman utama. Pengunjung mengisi kolom kata kunci, kemudian mengklik tombol "Cari". Sistem mengecek apakah kolom kata kunci sudah terisi. Jika sudah terisi maka sistem akan mencocokkan kolom kata kunci kemudian menampilkan hasil

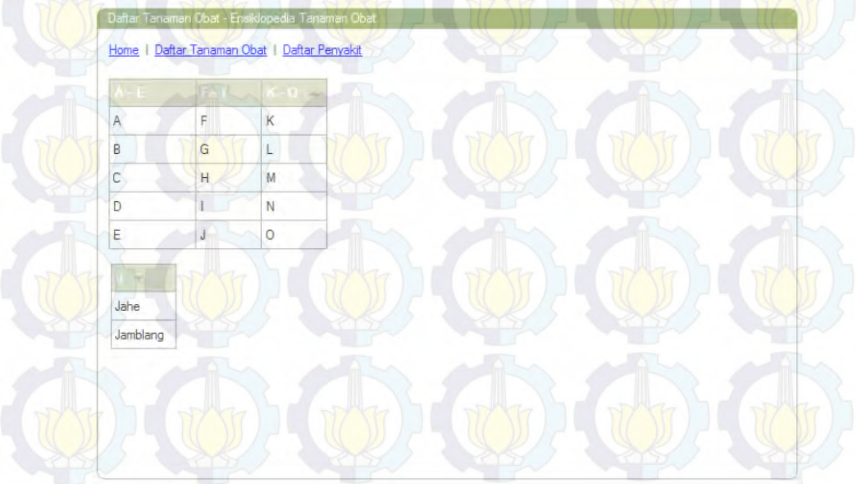
	pencarian.
Kolom kata kunci kosong	Sistem menampilkan pesan bahwa kolom kata kunci belum diisi dan harus diisi terlebih dahulu.
Tidak ada hasil yang cocok dengan kata kunci	Sistem menampilkan pesan bahwa tidak ada hasil yang sesuai.

Lampiran B

GUI Prototype



Gambar B. 1 Halaman Menu Utama



Gambar B. 2 Halaman Daftar Abjad Tanaman Obat

Jamblang - Ensiklopedia Tanaman Obat

[Home](#) | [Daftar Tanaman Obat](#) | [Daftar Penyakit](#)

Jamblang
 Kerajaan: Plantae
 Divisi: Magnoliophyta
 Kelas: Magnolopsida
 Ordo: Myrtales
 Famili: Myrtaceae
 Genus: Syzygium



Pohon yang kokoh, berkayu, diameter 10-30 m, berwarna putih kotor, dan tidak menggugurkan daun. [2] Kadang-kadang berbatang bengkok, tinggi hingga 20 m dan gemang mencapai 90 cm. Bercabang rendah dan bertajuk bulat atau tidak beraturan.

Daun-daunnya terletak berhadapan, bertangkai 1-3,5 cm. Helaian daun bundar telur terbalik agak jorong sampai jorong lonjong, 5-25 x 2-10 cm, pangkalnya lebar berbentuk pasak atau membundar.

Digunakan untuk obat : [Demam](#)

Tata Cara Penyajian : Untuk obat demam diperlukan sekitar 20 gram daun Jamblang yang segar, daun tersebut direbus dengan 1 gelas air selama 15 menit, setelah dingin disaring. Hasil saringan tersebut diminum sekaligus.

Gambar B. 3 Halaman Informasi Tanaman Obat

Daftar Penyakit - Ensiklopedia Tanaman Obat

[Home](#) | [Daftar Tanaman Obat](#) | [Daftar Penyakit](#)

A - E	F - I	K - O
A	F	K
B	G	L
C	H	M
D	I	N
E	J	O

U

Batuk

Bisul

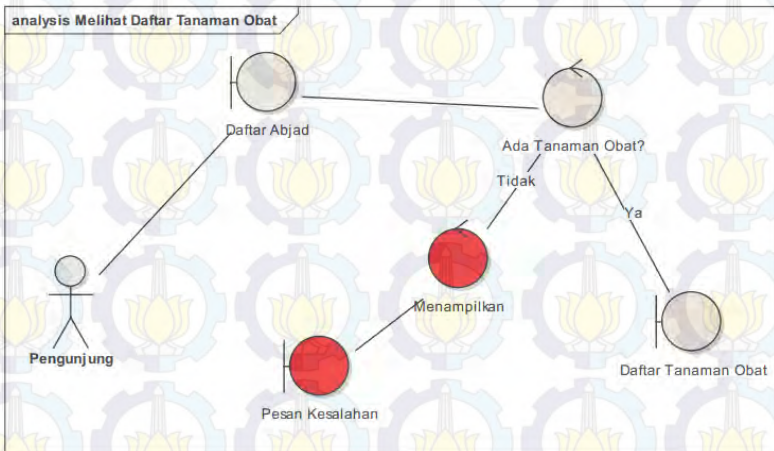
Gambar B. 4 Halaman Daftar Penyakit



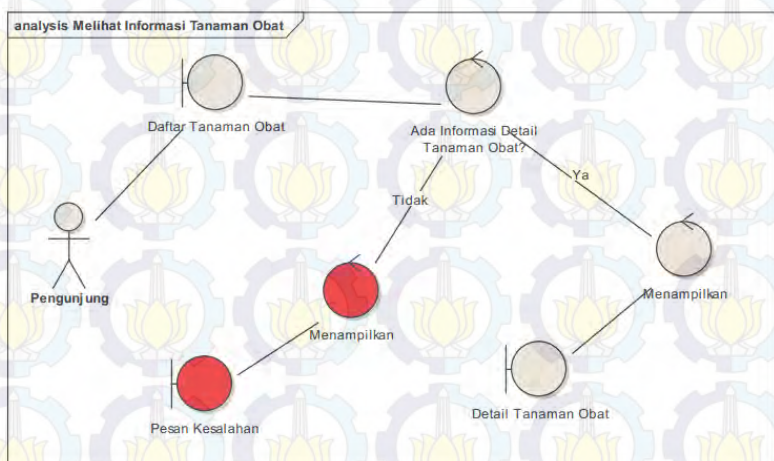
Gambar B. 5 Halaman Informasi Penyakit

Lampiran C

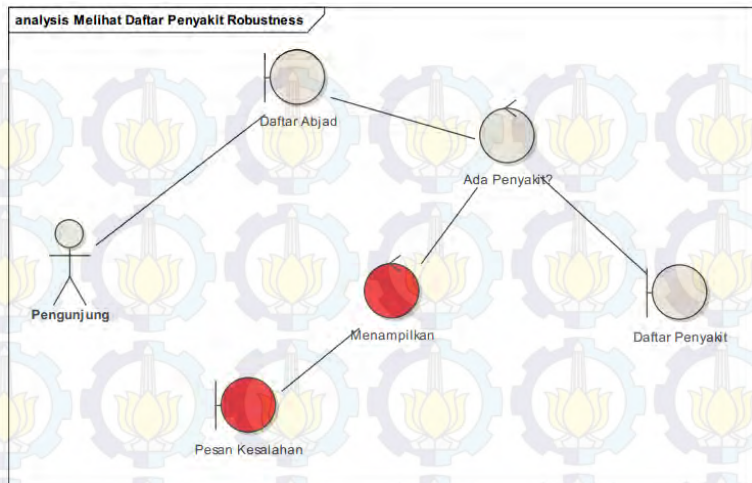
Robustness Diagram



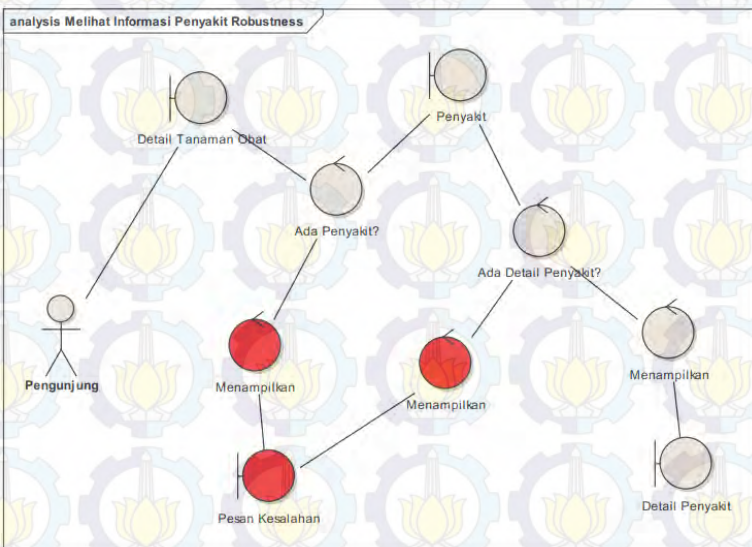
Gambar C. 1 Diagram Robustness Use Case Melihat Daftar Tanaman Obat



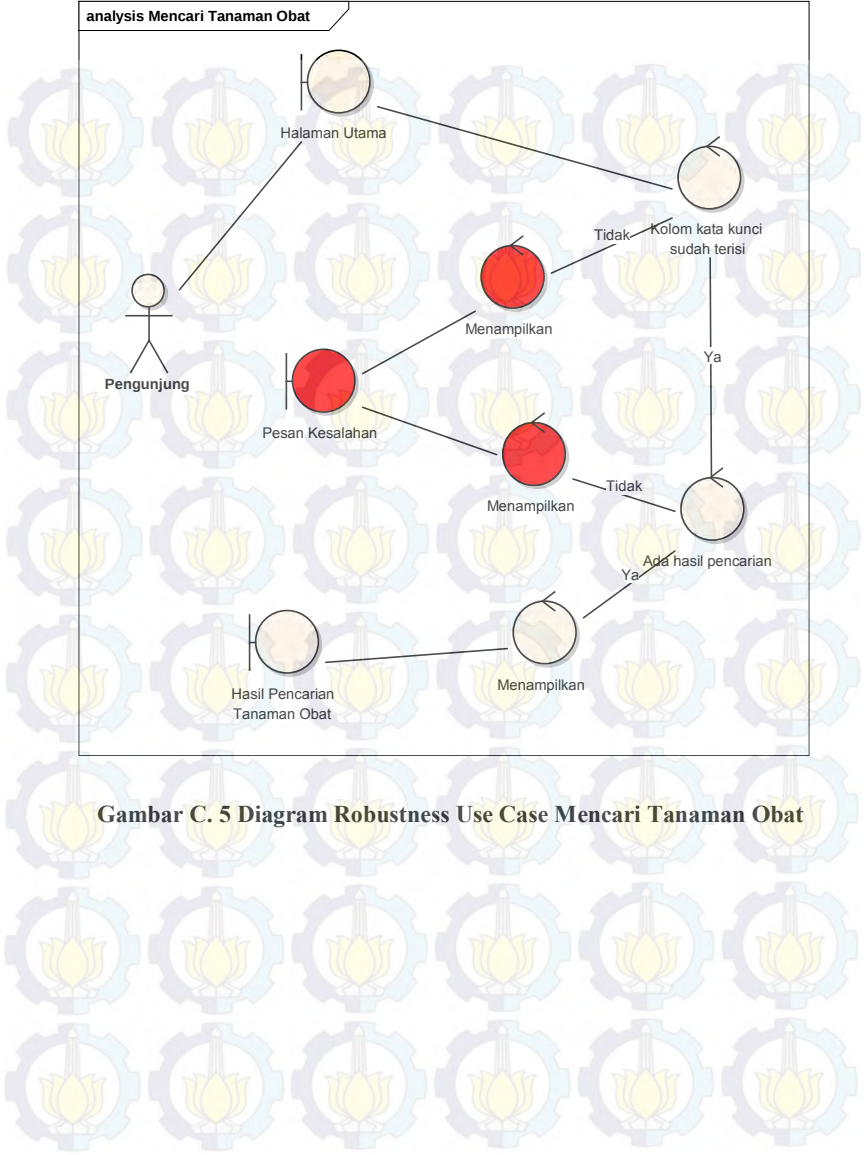
Gambar C. 2 Diagram Robustness Use Case Melihat Informasi Tanaman Obat



Gambar C. 3 Diagram Robustness Use Case Melihat Daftar Penyakit



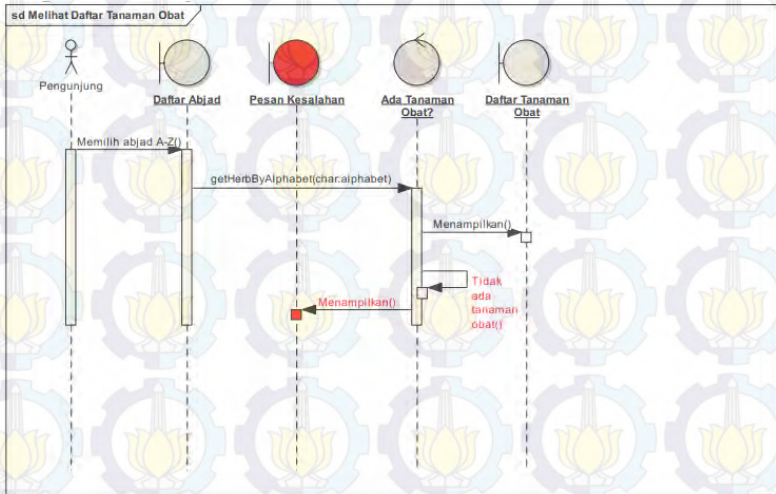
Gambar C. 4 Diagram Robustness Use Case Melihat Informasi Penyakit



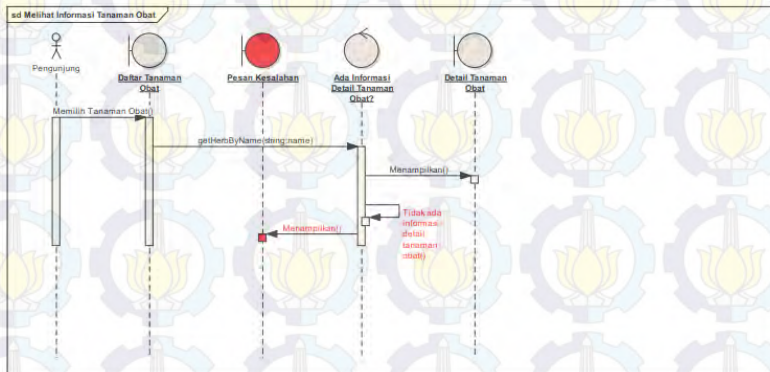
Gambar C. 5 Diagram Robustness Use Case Mencari Tanaman Obat

Lampiran D

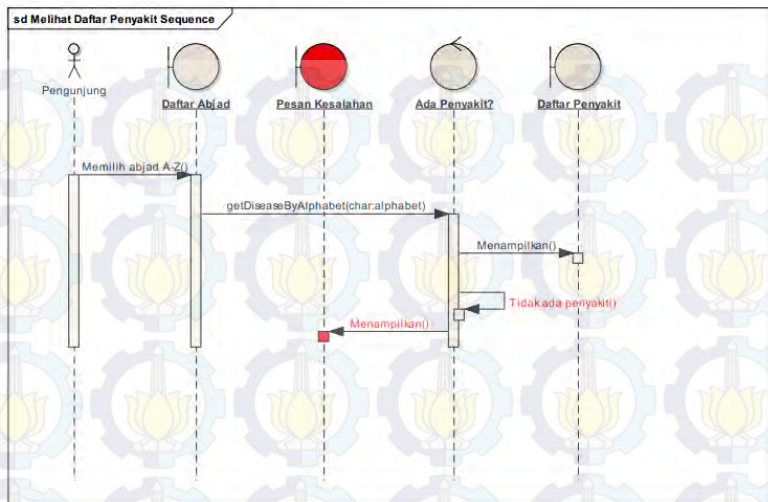
Sequence Diagram



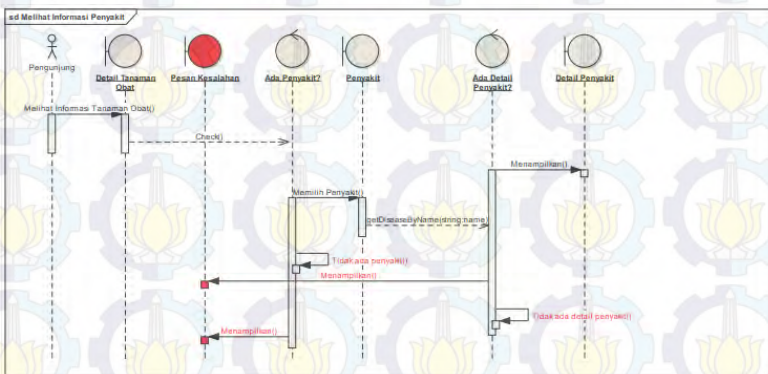
Gambar D. 1 Diagram Sequence Use Case Melihat Daftar Tanaman Obat



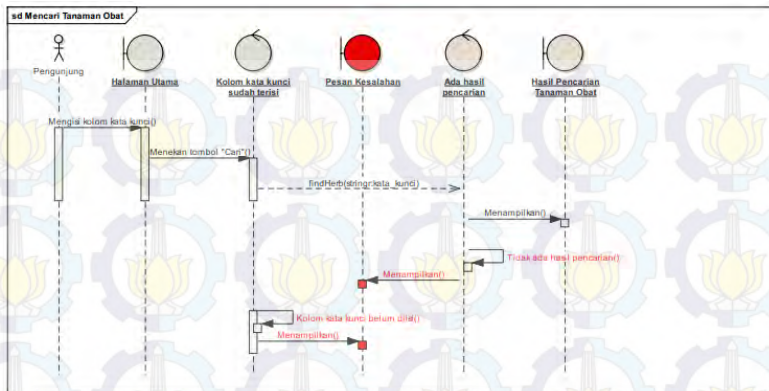
Gambar D. 2 Diagram Sequence Use Case Melihat Informasi Tanaman Obat



Gambar D. 3 Diagram Sequence Use Case Melihat Daftar Penyakit



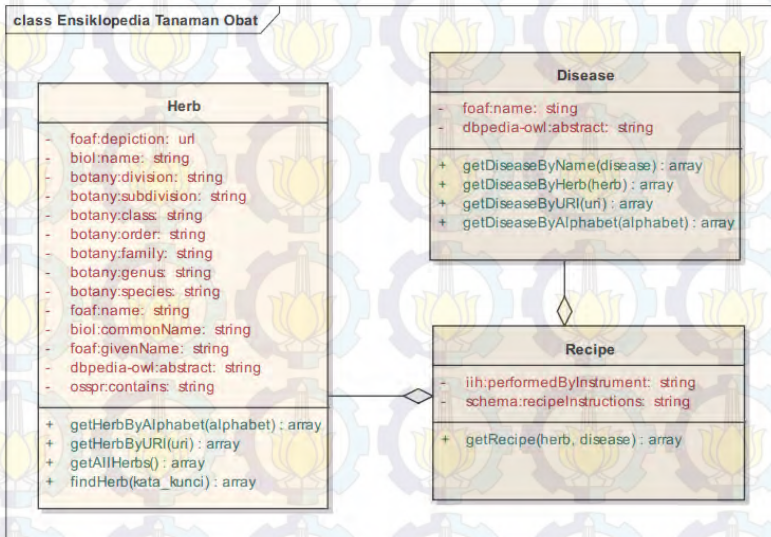
Gambar D. 4 Diagram Sequence Use Case Melihat Informasi Penyakit



Gambar D. 5 Diagram Sequence Use Case Mencari Tanaman Obat

Lampiran E

Class Diagram



Gambar E. 1 Class Diagram Ensiklopedia Tanaman Obat

Lampiran F

Test Case

1. Test Case Melihat Abjad Tanaman Obat

Tabel F. 1 Melihat Abjad Tanaman Obat

Test Case Code			TC-01		
Test Case Name			Melihat Abjad Tanaman Obat		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1	Pengguna Mengakses menu Daftar Tanaman Obat	-	Sistem menampilkan halaman daftar abjad.	Halaman menampilkan daftar abjad dari A-Z ditampilkan.	yang Sukses

2. Test Case Melihat Daftar Tanaman Obat

Tabel F. 2 Test Case Melihat Daftar Tanaman Obat

Test Case Code			TC-02		
Test Case Name			Melihat Daftar Tanaman Obat		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1	Pengguna Memilih Abjad	Abjad = "A"	Sistem menampilkan nama daftar tanaman obat yang dimulai dengan huruf "A"	Sistem menampilkan tanaman obat sebagai berikut: Adem Ati, Akasia, Ambre, Andong, Angsana, Anyelir, Apel, Apokad, Arbei, Aren, Asparagus, Awar-awar.	Sukses
2	Pengguna memilih Abjad	Abjad = "E"	Tanaman obat dengan huruf awalan E masih tidak ada, sehingga sistem menampilkan	Pesan peringatan bahwa tanaman obat dengan huruf awalan "E" tidak ada.	Sukses

			pesan tanaman dengan awalan tersebut tidak ada.	bahwa obat huruf “E”	
--	--	--	---	-------------------------------	--

3. Test Case Melihat Informasi Ilmiah Tanaman Obat

Tabel F. 3 Test Case Melihat Informasi Ilmiah Tanaman Obat

Test Case Code			TC-03		
Test Case Name			Melihat Informasi Ilmiah Tanaman Obat		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal

1	Pengguna memilih tanaman obat dari daftar tanaman obat.	Tanaman Obat = “Akasia”	Sistem menampilkan informasi ilmiah mengenai tanaman obat “Akasia”.	Informasi mengenai tanaman obat Akasia ditampilkan meliputi penjelasan mengenai nama ilmiah, abstrak, informasi kandungann kimia.	Sukses
---	---	-------------------------	---	---	--------

4. Test Case Melihat Daftar Penyakit Untuk Tiap Tanaman Obat

Tabel F. 4 Test Case Melihat Daftar Penyakit Untuk Tiap Tanaman Obat

Test Case Code			TC-04		
Test Case Name			Melihat Daftar Penyakit Untuk Tiap Tanaman Obat		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal

1	Pengguna memilih tanaman obat dari daftar tanaman obat.	Tanaman Obat = “Akasia”	Sistem menampilkan daftar penyakit yang dapat disembuhkan dengan tanaman obat “Akasia”.	Sistem menampilkan daftar penyakit yakni “Demam” yang dapat disembuhkan dengan “Akasia”.	Sukses
---	---	-------------------------	---	--	--------

5. Test Case Melihat Resep Obat Untuk Tiap Penyakit

Tabel F. 5 Test Case Melihat Resep Obat Untuk Tiap Penyakit

Test Case Code			TC-05		
Test Case Name			Melihat Resep Obat Untuk Tiap Penyakit		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal

1	Pengguna memilih tanaman obat dari daftar tanaman obat.	Tanaman Obat = “Akasia”	Sistem menampilkan penyakit yang dapat disembuhkan dengan tanaman obat “Akasia” beserta tata cara pengolahan resepnya.	Informasi mengenai tanaman obat Akasia. Penyakit yang dapat disembuhkan adalah “Demam”, dengan resep tata cara pengolahan tanaman obat tersebut.	Sukses
---	---	-------------------------	--	--	--------

6. Test Case Melihat Abjad Penyakit

Tabel F. 6 Test Case Melihat Abjad Penyakit

Test Case Code			TC-06		
Test Case Name			Melihat Abjad Penyakit		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1	Pengguna Mengakses		Sistem menampilkan	Halaman yang menampilkan daftar abjad	Sukses

	menu Daftar Penyakit		halaman daftar abjad.	dari A-Z ditampilkan.	
--	----------------------	--	-----------------------	-----------------------	--

7. Test Case Melihat Daftar Penyakit

Tabel F. 7 Test Case Melihat Daftar Penyakit

Test Case Code			TC-07		
Test Case Name			Melihat Daftar Penyakit		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1	Pengguna Memilih Abjad	Abjad = "A"	Sistem menampilkan nama daftar penyakit yang dimulai dengan huruf "A"	Sistem menampilkan tanaman obat sebagai berikut: Asma.	Sukses
2	Pengguna memilih	Abjad = "F"	Tanaman obat dengan huruf	Pesan peringatan bahwa tanaman obat dengan huruf	Sukses

	Abjad		awalan F masih tidak ada, sehingga sistem menampilkan pesan bahwa tanaman obat dengan huruf awalan “F” tersebut tidak ada	awalan “F” tidak ada.	
--	-------	--	--	-----------------------	--

8. Test Case Melihat Abstrak Penyakit

Tabel F. 8 Test Case Melihat Abstrak Penyakit

Test Case Code			TC-08		
Test Case Name			Melihat Abstrak Penyakit		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal

1	Pengguna Memilih Penyakit yang ada pada informasi tanaman obat.	Penyakit = “Demam”	Sistem menampilkan abstrak “Demam”.	Sistem menampilkan abstrak “Demam”.	Sukses
2	Pengguna Memilih Penyakit yang ada pada informasi tanaman obat.	Penyakit = “Kulit Gatal”	Jika informasi Kulit Gatal masih belum tersedia pada DBpedia, maka ada pesan peringatan bahwa abstrak mengenai penyakit tersebut belum tersedia.	Muncul pesan yang menerangkan bahwa abstrak penyakit Gatal belum tersedia	Sukses

9. Test Case Melihat Gambar Ilustrasi Penyakit

Tabel F. 9 Test Case Melihat Gambar Ilustrasi Penyakit

Test Case Code			TC-09		
Test Case Name			Melihat Gambar Ilustrasi Penyakit		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1	Pengguna Memilih Penyakit yang ada pada informasi tanaman obat.	Penyakit = “Demam”	Sistem menampilkan gambar ilustrasi “Demam”.	Sistem menampilkan gambar ilustrasi “Demam”.	Sukses

2	Pengguna Memilih Penyakit yang ada pada informasi tanaman obat.	Penyakit = “Kulit Gatal”	Jika gambar ilustrasi belum tersedia, maka akan ditampilkan gambar <i>default</i> yang menerangkan bahwa gambar belum tersedia.	Muncul gambar yang menerangkan bahwa gambar belum tersedia.	Sukses
---	---	--------------------------	---	---	--------

10. Test Case Melihat Daftar Tanaman Obat Untuk Tiap Penyakit

Tabel F. 10 Test Case Melihat Daftar Tanaman Obat Untuk Tiap Penyakit

Test Case Code	TC-10
Test Case Name	Melihat Daftar Tanaman Obat Untuk Tiap Penyakit

No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal
1	Pengguna Memilih Penyakit yang ada pada informasi tanaman obat.	Penyakit = "Demam"	Sistem menampilkan daftar Tanaman Obat yang dapat digunakan untuk obat penyakit "Demam".	Sistem daftar penyakit sebagai berikut: Akasia, Beluntas, Bawang Merah dan sebagainya sebanyak 24.	Sukses

11. Test Case Mencari Tanaman Obat

Tabel F. 11 Test Case Mencari Tanaman Obat

Test Case Code			TC-11		
Test Case Name			Mencari Tanaman Obat		
No	Aksi Tes	Data Tes	Hasil yang Diharapkan	Hasil Aktual	Sukses/gagal

1	Pengguna Mengakses Menu Utama	-	Sistem menampilkan halaman utama dengan kolom pencarian dan tombol “Search”.	Halaman utama dengan kolom pencarian dan tombol “Search” ditampilkan	Sukses
2	Pengguna melakukan pencarian.	Kata Kunci = “wasir”	Sistem menampilkan daftar hasil pencarian tanaman obat yang memiliki hubungan dengan wasir.	Sistem menampilkan hasil pencarian sebanyak 4 tanaman obat yaitu: Daun Duduk, Daun Ungu, Her, dan Sosor Bebek.	Sukses

3	Pengguna melakukan pencarian	Kata Kunci = "adasdaffa"	Jika sistem tidak menemukan tanaman obat yang berhubungan kata kunci tersebut, maka muncul pesan peringatan bahwa hasil tidak ditemukan.	Sistem menampilkan pesan peringatan bahwa tidak ditemukan hasil pencarian.	Sukses
---	------------------------------	--------------------------	--	--	--------

4	Pengguna melakukan pencarian.	Kata Kunci = null	Jika kata kunci belum diisi maka akan muncul pesan peringatan bahwa kata kunci belum diisi dan silahkan isi kolom kata kunci tersebut.	Sistem menampilkan pesan peringatan bahwa pengguna belum mengisi kata kunci dan diharuskan agar mengisi kata kunci.	Sukses
---	-------------------------------	-------------------	--	---	--------