



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - IF184802

# PERANCANGAN ONTOLOGI UNTUK PEMODELAN DATA PADA *VIRTUAL CHATBOT ASSISTANT* UNTUK STUDI KASUS APLIKASI GIFOOD.ID

WAHYU IVAN SATYAGRAHA  
NRP 05111540000155

Dosen Pembimbing  
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.  
Nurul Fajrin A., S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019









**TUGAS AKHIR - IF184802**

**PERANCANGAN ONTOLOGI UNTUK PEMODELAN  
DATA PADA *VIRTUAL CHATBOT ASSISTANT*  
UNTUK STUDI KASUS APLIKASI GIFOOD.ID**

**WAHYU IVAN SATYAGRAHA  
NRP 05111540000155**

**Dosen Pembimbing  
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.  
Nurul Fajrin A., S.Kom., M.Sc.**

**DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESIS - IF184802**

# **ONTOLOGY DESIGN FOR DATA MODELLING IN VIRTUAL CHATBOT ASSISTANT FOR GIFOOD.ID APPLICATION CASE STUDY**

**WAHYU IVAN SATYAGRAHA  
NRP 05111540000155**

**Supervisors**

**Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.  
Nurul Fajrin A., S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS  
Faculty of Information and Communication Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*[Halaman ini sengaja dikosongkan]*



**LEMBAR PENGESAHAN**  
**PERANCANGAN ONTOLOGI UNTUK**  
**PEMODELAN DATA PADA *VIRTUAL CHATBOT***  
**ASSISTANT UNTUK STUDI KASUS APLIKASI**  
**GIFOOD.ID**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Manajemen Informasi  
Program Studi S-1 Teknik Informatika  
Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:  
**WAHYU IVAN SATYAGRAHA**  
NRP : 0511154000155

Disetujui oleh Dosen Pembimbing **Tugas Akhir**

1. Dr.Tech. Ir. Raden Venantius Harto  
Ginardi, M.Sc.  
NIP: 196505181992031003  
(Pembimbing 1)
2. Nurul Fajrin Ariyani, S.kom.  
NIP: 198607222015042003  
(Pembimbing 2)

**SURABAYA**  
**JULI 2019**

*[HALAMAN INI SENGAJA DIKOSONGKAN]*

**PERANCANGAN ONTOLOGI UNTUK  
PEMODELAN DATA PADA *VIRTUAL CHATBOT  
ASSISTANT* UNTUK STUDI KASUS APLIKASI  
GIFOOD.ID**

Nama Mahasiswa : Wahyu Ivan Satyagraha  
NRP : 05111540000155  
Jurusan : Departemen Informatika FTIK-ITS  
Dosen Pembimbing 1 : Dr.Tech. Ir. Raden Venantius Hari  
Ginardi, M.Sc.  
Dosen Pembimbing 2 : Nurul Fajrin Ariyani, S.kom., M.Sc.

**ABSTRAK**

*Ontologi adalah sebuah cara untuk merepresentasikan pengetahuan dan konsep dalam sebuah domain informasi dengan cara memberikan makna, properti serta relasi pada konsep tersebut sehingga terhimpun dalam suatu domain pengetahuan dan membentuk sebuah basis pengetahuan. Pengaplikasian ontologi, sudah sangat luas dan mulai masuk ke banyak aspek pada kehidupan. Mulai dari digunakan untuk basis data, digabung dengan chatbot, sampai untuk fraud detection.*

*Membangun sebuah ontologi diperlukan suatu analisis dan perancangan yang matang. Bagaimana dalam menentukan kelas, property terkait, hingga individu yang ada di dalamnya. Selain itu, ontologi juga harus sesuai dengan studi kasus yang ada. Diperlukan juga eksplorasi lebih lanjut dalam membangun sebuah ontologi*

*Pada tugas akhir ini, ontologi akan dibangun untuk studi kasus aplikasi gifood.id. ontologi dibangun dengan menggunakan Bahasa OWL (Ontology Web Language), dan dengan menggunakan tools pembantu Protégé.*

*Untuk mengakses ontologi yang sudah dibangun, akan dibangun juga sebuah sistem berbasis web yang nantinya akan terhubung dengan ontologi yang sudah dibangun. Menggunakan beberapa Library untuk dapat mengakses ontologi dari sistem, dan Bahasa pemrograman HTML, CSS, Javascript, PHP, Python.*

***Kata kunci: analisis, ontologi, protege, web***

# **ONTOLOGY DESIGN FOR DATA MODELLING ON VIRTUAL CHATBOT ASSISTANT GIFOOD.ID APPLICATION CASE STUDY**

Name : Wahyu Ivan Satyagraha  
NRP : 05111540000155  
Major : Informatics Department FTIK-ITS  
Supervisor I : Dr.Tech. Ir. Raden Venantius Hari  
Ginardi, M.Sc.  
Supervisor II : Nurul Fajrin Ariyani, S.kom., M.Sc.

## **ABSTRACT**

*Ontology is a way to represent knowledge and concepts in a domain of information by giving meaning, property and relations to the concept so that it is gathered in a domain of knowledge and forms a knowledge base. The application of ontology, has been very broad and began to enter into many aspects of life. Starting from being used for databases, combined with chatbots, to fraud detection.*

*Building an ontology requires careful analysis and design. How to determine classes, related properties, to individuals in them. In addition, ontology must also be in accordance with existing case studies. Further exploration is needed in building an ontology*

*In this final project, ontology will be built for case studies of gifood.id applications. ontology is built using the OWL Language (Ontology Web Language), and by using Protégé's help tools.*

*To access the ontology that has been built, a web-based system will be built which will be connected to the ontology that has been built. Use several libraries to be able*

*to access the ontology of the system, and HTML, CSS, Javascript, PHP, Python programming languages.*

***Keywords: analysis, ontology, protege, web***

## **KATA PENGANTAR**

Alhamdulillahirobbil ‘alamiin, puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **PERANCANGAN ONTOLOGI UNTUK PEMODELAN DATA PADA *VIRTUAL CHATBOT ASSISTANT* UNTUK STUDI KASUS APLIKASI *GIFOOD.ID***

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini serta senantiasa menemani dan memberi kemudahan kepada penulis dalam menyelesaikan Tugas Akhir.
2. Kedua Orang Tua Bapak Rahmat Broto Triaji dan Ibu Susi Ernawati, serta Adik Nabila Dyah Amelia yang telah memberikan doa dan dukungan selama ini.
3. Bapak Dr.Tech. Ir. Raden Venantius Hari Ginardi, M.Sc. selaku dosen pembimbing I yang selalu memberikan motivasi dan mengarahkan penulis ke arah yang benar selama pengerjaan Tugas Akhir.
4. Ibu Nurul Fajrin Ariyani, S.kom., M.Sc. selaku dosen pembimbing II yang senantiasa memberikan masukan, arahan, dan bantuan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
5. Bapak dan Ibu dosen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
6. Om Irfan Hidayat, S.Si., M.Kom yang sudah memberikan inspirasi dan masukan sehingga penulis dapat menyelesaikan Tugas Akhir ini.

7. Pihak Gifood, Mas Fathin Naufal dan Mas Rizqy Hidayat yang sudah memberikan banyak bantuan dalam pengerjaan Tugas Akhir ini.
8. Khairunnisa` Rahma Mardiyani, yang telah memberikan semangat dan bantuan kepada penulis selama proses pengerjaan Tugas Akhir ini.
9. Satriyo Nugroho, teman seperjuangan Tugas Akhir yang selama pengerjaan Tugas Akhir saling membantu dan berkoordinasi demi kesuksesan Tugas Akhir ini.
10. Teman-teman lab Manajemen Informasi yang saling menyemangati dan membantu selama mengerjakan Tugas Akhir di Lab Manajemen Informasi.
11. Gerald, Subhan, Dio, Glenn, Eko, Fikri, Ufa, Wahyu Nugraha, Unggul dan Daffa yang selalu memberikan warna dan hiburan selama kuliah di Departemen Informatika ITS.
12. Teman-teman Badan Pengurus Harian (BPH) Schematics 2017 yang telah memberikan kesempatan penulis untuk mendapatkan pengalaman lebih selama 1 tahun kepengurusan.
13. Teman-teman angkatan 2015 yang telah memberikan motivasi dan semangat selama penulis berkuliah di Informatika ITS.
14. Serta pihak lain yang namanya tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih memiliki kekurangan. Oleh karena itu, dengan tangan terbuka, penulis menerima segala saran dan kritik dari pembaca untuk perbaikan ke depannya.

Surabaya, Juli 2019

Wahyu Ivan Satyagraha





## DAFTAR ISI

LEMBAR PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI .....	xvi
DAFTAR GAMBAR.....	xx
DAFTAR TABEL .....	xxiii
DAFTAR KODE SUMBER.....	xxiv
1 BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	4
1.4 Tujuan.....	4
1.5 Manfaat .....	5
1.6 Metodologi Pembuatan Tugas Akhir.....	5
1.7 Sistematika Penulisan .....	7
2 BAB II TINJAUAN PUSTAKA .....	9
2.1 Ontologi .....	9
2.2 Pemodelan.....	12
2.3 Protégé .....	12
2.4 OWL .....	13
2.5 SPARQL .....	16
2.6 RDF.....	18
2.7 EasyRDF.....	20
2.8 RDFlib .....	21
2.9 PHP .....	21
2.10 Python.....	23
3 BAB III ANALISIS DAN PERANCANGAN .....	25
3.1 Analisis Dataset .....	25
3.2 Analisis dan Perancangan Ontologi .....	26
3.2.1 Deskripsi Kelas .....	33
3.2.2 Deskripsi <i>Object Properties</i> .....	35

3.2.3	Deskripsi <i>Data Properties</i> .....	38
3.2.4	Deskripsi <i>Instance</i> .....	39
3.2.5	Deskripsi Anotasi .....	40
3.2.6	Deskripsi <i>Rule</i> .....	42
3.2.7	Skenario Uji Coba .....	42
3.3	Analisis dan Perancangan Sistem .....	51
3.3.1	Deskripsi Umum Sistem .....	51
4	BAB IV IMPLEMENTASI .....	57
4.1	Lingkungan Implementasi Perangkat Lunak .....	57
4.2	Implementasi .....	57
4.2.1	Implementasi Fitur Melihat <i>Instances</i> Pada Ontologi .....	58
4.2.2	Implementasi Fitur Melihat Studi Kasus .....	60
4.2.3	Implementasi Fitur Menambah <i>Instance</i> .....	62
4.2.4	Implementasi Fitur Menghapus <i>Instance</i> dari Ontologi .....	64
5	BAB V PENGUJIAN DAN EVALUASI .....	67
5.1	Lingkungan Pengujian .....	67
5.2	Pengujian Fitur .....	67
5.2.1	Fitur Melihat <i>Instance</i> dari Ontologi .....	68
5.2.2	Fitur Melihat Studi Kasus .....	69
5.2.3	Fitur Menambah <i>Instance</i> ke Dalam Ontologi .....	79
5.2.4	Fitur Menghapus <i>Instance</i> dari Ontologi .....	81
5.3	Evaluasi Pengujian Fitur .....	83
5.4	Perbandingan Pengujian Fitur dengan Protégé .....	83
5.4.1	Fitur Melihat <i>Instance</i> dari Ontologi .....	84
5.4.2	Fitur Menambah <i>Instance</i> ke Dalam Ontologi .....	86
5.4.3	Fitur Menghapus <i>Instance</i> dari Ontologi .....	88
6	BAB VI KESIMPULAN DAN SARAN .....	91
6.1	Kesimpulan .....	91
6.2	Saran .....	92
7	DAFTAR PUSTAKA .....	93
8	LAMPIRAN A .....	95
9	LAMPIRAN B .....	102
10	LAMPIRAN C .....	103
11	LAMPIRAN D .....	104
12	LAMPIRAN E .....	105

13	LAMPIRAN F .....	106
14	LAMPIRAN G .....	110
15	LAMPIRAN H .....	111
16	LAMPIRAN I.....	112
17	LAMPIRAN J.....	113
18	LAMPIRAN K .....	114
19	BIODATA PENULIS.....	116

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Diagram Siklus Perancangan Ontologi.....	12
Gambar 3.1 Alur Perancangan Ontologi .....	28
Gambar 3.2 Diagram rancangan ontologi.....	29
Gambar 3.3 Contoh Hubungan untuk Instance “Sate Ayam” .....	32
Gambar 3.4 Instance yang Teribat pada Contoh “Sate Ayam” ...	33
Gambar 3.5 Hierarki Kelas pada Ontologi .....	34
Gambar 4.1 Halaman Implementasi Fitur Melihat <i>Instance</i> .....	58
Gambar 4.2 Halaman Fitur Melihat Studi Kasus .....	60
Gambar 4.3 Form Untuk Mengisi <i>Instance</i> Baru .....	63
Gambar 5.1 Hasil Pengujian Fitur Melihat <i>Instance</i> .....	68
Gambar 5.2 Hasil Pengujian Fitur Studi Kasus nomor 1.....	70
Gambar 5.3 Hasil Pengujian Fitur Studi Kasus nomor 2.....	70
Gambar 5.4 Hasil Pengujian Fitur Studi Kasus nomor 3.....	71
Gambar 5.5 Hasil Pengujian Fitur Studi Kasus nomor 4.....	71
Gambar 5.6 Hasil Pengujian Fitur Studi Kasus nomor 5.....	72
Gambar 5.7 Hasil Pengujian Fitur Studi Kasus nomor 6.....	73
Gambar 5.8 Hasil Pengujian Fitur Studi Kasus nomor 7.....	73
Gambar 5.9 Hasil Pengujian Fitur Studi Kasus nomor 8.....	74
Gambar 5.10 Hasil Pengujian Fitur Studi Kasus nomor 9.....	74
Gambar 5.11 Hasil Pengujian Fitur Studi Kasus nomor 10.....	75
Gambar 5.12 Hasil Pengujian Fitur Studi Kasus nomor S1 .....	75
Gambar 5.13 Hasil Pengujian Fitur Studi Kasus nomor S2 .....	76
Gambar 5.14 Hasil Pengujian Fitur Studi Kasus nomor S3 .....	76
Gambar 5.15 Hasil Pengujian Fitur Studi Kasus nomor S4 .....	77
Gambar 5.16 Hasil Pengujian Fitur Studi Kasus nomor S5 .....	77
Gambar 5.17 Hasil Pengujian Fitur Menambah <i>Instance</i> .....	80
Gambar 5.18 Hasil Pengujian Fitur Menambah <i>Instance</i> .....	81
Gambar 5.19 Hasil Pengujian Fitur Menghapus <i>Instance</i> .....	82
Gambar 5.20 Hasil Pengujian Fitur Menghapus <i>Instance</i> .....	82
Gambar 5.21 Hasil Pembacaan dari Protégé untuk fitur Melihat <i>Instance</i> dari Ontologi untuk kelas Foody.....	85
Gambar 5.22 Hasil dari Pengujian pada Protégé untuk fitur Melihat <i>Instance</i> dari Ontologi untuk kelas Bahan.....	86

Gambar 5.23 Hasil Pembacaan dari Protégé untuk Fitur Menambah <i>Instance</i> ke Dalam Ontologi.....	87
Gambar 5.24 Hasil Pembacaan dari Protégé untuk Fitur Menambah <i>Instance</i> ke Dalam Ontologi.....	87
Gambar 5.25 Hasil Pembacaan dari Protégé untuk Fitur Menghapus <i>Instance</i> dari Dalam Ontologi.....	89

*[Halaman ini sengaja dikosongkan]*



## DAFTAR TABEL

Tabel 3.1 <i>Literature Review</i> Penelitian Terdahulu .....	26
Tabel 3.2 Deskripsi Kelas .....	34
Tabel 3.3 Deskripsi <i>Object Properties</i> .....	35
Tabel 3.4 Deskripsi <i>Data Properties</i> .....	38
Tabel 3.5 Deskripsi Instance .....	39
Tabel 3.6 Beberapa contoh <i>instance</i> Ditambahkan Anotasi.....	41
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	57
Tabel 5.1 Lingkungan Pengujian.....	67
Tabel 5.2 Pengujian Fitur Melihat <i>Instance</i> dari Ontologi .....	69
Tabel 5.3 Pengujian Fitur Melihat Studi Kasus.....	78
Tabel 5.4 Pengujian Fitur Menambah <i>Instance</i> ke Ontologi.....	80
Tabel 5.5 Pengujian Fitur Menghapus <i>Instance</i> dari Ontologi....	83
Tabel 5.6 Hasil Evaluasi Pengujian Fitur .....	83

## DAFTAR KODE SUMBER

Kode Sumber 2.1 Contoh Penggalan Sintaks dari SPARQL .....	18
Kode Sumber 2.2 Contoh Sintaks dari RDF.....	19
Kode Sumber 3.1 Prefix yang digunakan pada <i>query</i> .....	43
Kode Sumber 3.2 Sintaks SPARQL studi kasus 1 .....	44
Kode Sumber 3.3 Sintaks SPARQL studi kasus 2 .....	44
Kode Sumber 3.4 Sintaks SPARQL studi kasus 3 .....	45
Kode Sumber 3.5 Sintaks SPARQL studi kasus 4 .....	45
Kode Sumber 3.6 Sintaks SPARQL studi kasus 5 .....	46
Kode Sumber 3.7 Sintaks SPARQL studi kasus 6 .....	46
Kode Sumber 3.8 Sintaks SPARQL studi kasus 7 .....	47
Kode Sumber 3.9 Sintaks SPARQL studi kasus 8 .....	47
Kode Sumber 3.10 Sintaks SPARQL studi kasus 9 .....	48
Kode Sumber 3.11 Sintaks SPARQL studi kasus 10 .....	48
Kode Sumber 3.12 Sintaks SPARQL studi kasus S1 .....	49
Kode Sumber 3.13 Sintaks SPARQL studi kasus S2 .....	50
Kode Sumber 3.14 Sintaks SPARQL studi kasus S3 .....	50
Kode Sumber 3.15 Sintaks SPARQL studi kasus S4 .....	51
Kode Sumber 3.16 Sintaks SPARQL studi kasus S5 .....	51
Kode Sumber 4.1 Dropdown Untuk Kelas Foody.....	59
Kode Sumber 4.2 Dropdown Untuk Kelas Bahan.....	59
Kode Sumber 4.3 Script PHP Melihat Studi Kasus.....	61
Kode Sumber 4.4 Kode sumber dari Script Python untuk Implementasi Fitur Melihat Studi Kasus untuk query nomor 1 ..	62
Kode Sumber 4.5 Script PHP untuk Fitur Menambah <i>Instance</i> ..	63
Kode Sumber 4.6 Script Python untuk Menambah <i>Instance</i> .....	64
Kode Sumber 4.7 Script PHP untuk Menghapus <i>Instance</i> .....	65
Kode Sumber 4.8 Script Python untuk Menghapus <i>Instance</i> .....	65

*[Halaman ini sengaja dikosongkan]*



# BAB I

## PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### 1.1 Latar Belakang

Saat ini, perkembangan *Artificial Intelligence* atau yang biasa disebut dengan AI mulai banyak dimanfaatkan dalam kehidupan sehari-hari. Dari banyak penerapan AI dalam kehidupan sehari-hari ada satu yang saat ini sedang banyak dilirik, yaitu *ChatBot*. Saat ini, kemampuan *ChatBot* dalam merespon *user* sangat bergantung kepada data yang berada di belakang nya. *Chatbot* biasanya menggunakan data yang sudah memiliki pola yang terprediksi. Sehingga sejak awal, *Programmer* sudah menentukan *Chatbot* ini bisa berbicara seperti apa, dan menjawab kepada *user* seperti apa. Hal ini berimbas kepada kesan *ChatBot* yang kaku. *User* akan langsung menyadari bahwa mereka sedang berinteraksi bukan dengan manusia sebenarnya. Saat ini, untuk memperbaiki keterbatasan chatbot seperti yang sudah dijelaskan diatas, sudah cukup banyak sumber yang menjelaskan solusi nya. Salah satu nya adalah mengimplementasikan ontologi ke dalam chatbot (*Ontology Based Chatbot*)<sup>1</sup>.

Berbicara mengenai ontologi, saat ini implementasi ontologi sudah cukup banyak dan cukup banyak membantu kehidupan manusia. Contohnya adalah implementasi ontologi pada chatbot, dapat dijadikan basis data pada suatu sistem pakar,

---

<sup>1</sup> A. Masychef, "Smart and Smarter : How Structured Content and Ontology Enrich User Experience with Chatbots", diakses dari <http://intuillion.com/2017/12/12/smart-and-smarter-how-structured-content-and-ontology-enrich-user-experience-with-chatbots/>, pada tanggal 28 December 2018.

*fraud detection*, dan dapat diintegrasikan dengan sistem yang sudah ada. Ontologi memiliki fitur *reasoning* (penalaran), dimana ini adalah salah satu fitur unggulan pada ontologi. Reasoning adalah kemampuan ontologi untuk menghasilkan fakta-fakta yang bersifat eksplisit. Ontologi juga memiliki apa yang disebut sebagai *relationship* (relasi). Dengan relasi, maka ontologi menjadi lebih dinamis. Kedua hal ini lah yang menjadikan ontologi, saat ini menjadi suatu hal yang sangat menarik<sup>2</sup>.

Tugas akhir ini mencoba untuk memperbaiki, dan menyelesaikan salah satu masalah pada studi kasus, yaitu pada aplikasi *gifood.id*. *gifood.id* adalah sebuah platform yang menghubungkan orang-orang yang memiliki makanan berlebih dengan mereka yang membutuhkan nya<sup>3</sup>. Dengan kata lain, mereka fokus pada pemberantasan *food waste*. Melalui sistem mereka, *food waste* berhasil ditekan. Namun perilaku *user gifood.id* yang kurang menghargai makanan yang mereka ambil, bisa membuat tujuan mereka, yaitu memberantas *food waste* bisa saja menjadi sia-sia. Sebagai contoh, ada *user* yang menemukan gudeg dalam daftar makanan yang tersedia. Bisa jadi dia tertarik dengan makanan tersebut namun tidak mengetahui dengan pasti apa saja kandungan bahan yang ada di dalam makanan gudeg. Berbekal rasa lapar dan rasa penasaran nya, *user* itu pun akhirnya mengambil makanan tersebut. Ternyata setelah diambil, *user* tersebut tidak menyukainya. Tidak mungkin makanan tersebut dikembalikan ke *gifood*. Maka, sudah pasti makanan tersebut akan dibuang kembali oleh *user*.

Kesimpulan nya. walaupun *gifood* sudah menyalurkan makanan kepada yang membutuhkan tetapi bila pihak yang membutuhkan tidak bertanggung jawab, seperti ternyata seperti yang sudah dijabarkan diatas, tidak cocok terhadap makanan yang

---

<sup>2</sup> Ontotext, "What are Ontologies?", diakses dari <https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/>, pada tanggal 23 June 2019.

<sup>3</sup> Gifood, "Gifood", diakses dari <https://gifood.id/> pada tanggal 29 December 2018.

diambil, porsi makanan yang diambil ternyata terlalu besar, atau pihak yang membutuhkan ternyata mempunyai suatu alergi tertentu yang pihak tersebut tidak mengetahui dan baru muncul ketika sedang mengkonsumsi makanan tertentu. Maka biasanya, makanan tersebut oleh pihak terkait akan kembali dibuang. Tentu ini akan menjadi *food waste*, dan ini bukan lah suatu masalah yang berada pada domain gifood.id.

Melihat permasalahan diatas, solusi yang ditawarkan pada tugas akhir ini adalah membuat sebuah sistem untuk memberikan informasi kepada *user* gifood mengenai makanan yang mereka ambil. Informasi tersebut bisa berupa nama makanan, cara penyajiannya seperti apa, dan kandungan bahan pada suatu makanan tersebut. Sistem ini menggunakan ontologi sebagai basis data nya, sehingga kedepannya akan ada eksplorasi lebih jauh lagi mengenai ontologi.

Hasil yang diharapkan dari pengerjaan Tugas Akhir ini adalah kedepannya, ontologi dapat dirancang dengan baik sehingga nantinya, dapat diimplementasikan ke dalam sistem yang akan dibuat. Pengeksplorasian ontologi lebih jauh selama pengerjaan tugas akhir ini akan membuat pemahaman mengenai ontologi semakin dalam. Pengimplementasian ontologi ini juga tidak hanya dibatasi hanya kepada sistem ini saja. Namun rencananya juga bisa diimplementasikan ke dalam aplikasi lainnya selama studi kasusnya sama. Misal chatbot dengan studi kasus yang sama, atau integrasi langsung ke dalam sistem utama dari gifood.id.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana cara merancang dan membangun sebuah ontologi untuk studi kasus aplikasi gifood?
2. Bagaimana cara mengimplementasikan ontologi untuk studi kasus aplikasi gifood ke dalam sebuah sistem?

3. Bagaimana cara mengembangkan sebuah sistem yang dapat mengakses ontologi untuk studi kasus aplikasi gifood?
4. Apa saja data yang harus dimodelkan ke dalam ontologi untuk studi kasus aplikasi gifood?
5. Bagaimana cara menguji kebenaran dari ontologi untuk studi kasus aplikasi gifood?

### 1.3 Batasan Masalah

Batasan Masalah pada Tugas Akhir ini antara lain :

1. Ontologi akan dibangun dari nol dan tidak melibatkan ontologi lain.
2. Ontologi dibangun dengan menggunakan Bahasa *Ontology Web Language* (OWL).
3. Ontologi dibangun untuk sistem informasi dengan studi kasus aplikasi gifood.
4. Bahasa yang digunakan dalam ontologi adalah Bahasa Indonesia.
5. Ontologi hanya memodelkan informasi seputar studi kasus aplikasi gifood.
6. Ontologi dibangun dengan menggunakan *Tools* pembantu Protégé.

### 1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain adalah sebagai berikut :

1. Memodelkan data ke dalam Ontologi
2. Membangun Ontologi, dan mengimplementasikan nya ke dalam sebuah aplikasi, dalam tugas akhir ini, sistem informasi berbasis web



3. Mencoba mengeksplorasi lebih jauh mengenai ontologi, apa yang dapat dilakukan dengan ontologi, bagaimana merancang ontologi yang baik dan benar.

## 1.5 Manfaat

Manfaat dari tugas akhir ini adalah diharapkan, dapat memperbaiki perilaku pihak yang membutuhkan pada aplikasi gifood agar kedepannya lebih bijak dalam mengambil makanan, dan mengajag agar bersama-sama memberantas *food waste*.

## 1.6 Metodologi Pembuatan Tugas Akhir

Adapun beberapa tahap dalam proses pengerjaan Tugas Akhir ini, yaitu:

1. Penyusunan proposal Tugas Akhir

Tahap pertama dalam proses pengerjaan Tugas Akhir ini adalah menyusun proposal Tugas Akhir. Pada proposal Tugas Akhir ini diajukan Perancangan Ontologi untuk pemodelan data untuk *Virtual Chatbot Assistant* yang nantinya diperuntukkan untuk aplikasi gifood.id.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam merancang serta membangun ontologi yaitu mengenai ontologi itu sendiri, teknologi yang berada di belakangnya, pengetahuan mengenai tools pembantu yang dapat digunakan, serta pengetahuan untuk mengembangkan aplikasi akhirnya.

3. Analisis dan perancangan

Tahap ini meliputi perumusan dan perancangan untuk ontologi dan sistem yang akan dibangun. Disini juga akan dirumuskan studi kasus yang diperuntukkan untuk menguji ontologi yang akan dibangun

#### 4. Implementasi

Implementasi pembangunan Ontologi menggunakan Bahasa OWL dan dibantu dengan tools pembantu protégé. Disini juga akan diimplementasikan sistem informasi berbasis web untuk mengakses ontologi. Selanjutnya adalah bagaimana mengimplementasikan agar nantinya sistem dan ontologi dapat terhubung. Penggunaan beberapa *Library* akan diperlukan disini.

#### 5. Pengujian dan evaluasi

Pengujian akan dilakukan dengan melakukan pengaksesan dan manipulasi ontologi dari sistem. Selain itu, terdapat juga pengujian ontologi dengan menggunakan studi kasus. Apakah *Output* dari sistem akan sesuai dengan yang diminta dengan studi kasus ataukah berbeda. Pengujian diakhiri dengan membandingkan *Output* yang dihasilkan pada sistem dengan *Output* yang dihasilkan pada tools pembantu protégé. Setelah itu akan ditarik evaluasi dari rangkaian pengujian

#### 6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini. Pada tahap ini juga disertakan hasil dari implementasi perangkat lunak yang telah dibuat. Sistematika penulisan buku Tugas Akhir ini secara garis besar antara lain:

1. Pendahuluan
  - a. Latar Belakang
  - b. Rumusan Masalah
  - c. Batasan Masalah
  - d. Tujuan
  - e. Manfaat
  - f. Metodologi Pembuatan Tugas Akhir
  - g. Sistematika Penulisan
2. Tinjauan Pustaka
3. Analisis dan Perancangan Sistem
4. Implementasi
5. Pengujian dan Evaluasi

6. Kesimpulan dan Saran
7. Daftar Pustaka

## **1.7 Sistematika Penulisan**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

### **Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

### **Bab II Tinjauan Pustaka**

Memaparkan dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

### **Bab III Analisis dan Perancangan Sistem**

Membahas tentang analisis permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, lingkungan perancangan, perancangan sistem, dan struktur data.

### **Bab IV Implementasi**

Bab ini berisi implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

### **Bab V Pengujian dan Evaluasi**

Bab ini membahas kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja

dari sistem yang telah dibuat.

### **Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan pada Tugas Akhir ini. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

### **Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

### **Lampiran**

Merupakan bab tambahan.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Ontologi**

Dalam ilmu komputer dan ilmu informasi, ontologi mencakup representasi, penamaan formal, definisi kategori, properti, hubungan antara konsep, data, dan entitas yang mendukung satu, banyak, atau semua domain. Setiap bidang menciptakan ontologi untuk membatasi kerumitan dan mengatur informasi menjadi data dan pengetahuan. Saat ontologi baru dibuat, penggunaannya diharapkan meningkatkan penyelesaian masalah dalam domain tersebut. Ontologi menurut manusia adalah mudah dibaca, komprehensif, bisa dibagi dan formal yang artinya bahwa mereka diekspresikan dalam bahasa yang telah didefinisikan dengan baik secara semantik [1].

Ontologi pertama kali dibuat pada tahun 1990. Pada waktu itu, belum ada metodologi atau petunjuk untuk mempermudah pembuatan ontologi. Setelah beberapa pengalaman di lapangan, Gruber memperkenalkan prinsip untuk mendesain ontologi. Apa yang dilakukan Gruber, adalah yang pertama kali untuk mendeskripsikan peran ontologi dalam aktivitas berbagi pengetahuan, dan menjadi patokan petunjuk untuk pengembangan ontologi selanjutnya. Pengembangan ontologi menjadi jauh lebih jelas dengan pengembangan ontologi lainnya [2].

Sejak *Google* memulai sebuah inisiatif yang disebut Graf Pengetahuan, sejumlah besar penelitian telah menggunakan graf pengetahuan frase sebagai istilah umum. Meskipun tidak ada

definisi yang jelas untuk istilah graf pengetahuan, kadang-kadang digunakan sebagai sinonim untuk ontologi. Salah satu interpretasi yang umum adalah bahwa graf pengetahuan mewakili kumpulan deskripsi entitas yang saling terkait - objek dunia nyata, peristiwa, situasi atau konsep abstrak<sup>4</sup>.

Ontologi memiliki banyak kesamaan struktural, terlepas dari bahasa yang digunakan untuk mengekspresikannya. Kebanyakan ontologi menggambarkan individu (*instances*), kelas (konsep), atribut, dan hubungan. Komponen ontologi yang umum meliputi:

1. Individu
2. Kelas
3. Atribut
4. Hubungan/relasi
5. Fungsi
6. Batasan
7. Aturan
8. Aksioma

Banyak disiplin ilmu sekarang mengembangkan ontologi terstandarisasi yang dapat digunakan oleh para ahli domain untuk berbagi dan membuat anotasi informasi di bidangnya. Kedokteran, misalnya, telah menghasilkan kosa kata yang besar, terstandarisasi, terstruktur seperti SNOMED (Price dan Spackman 2000) dan jaringan semantik Sistem Bahasa Medis Bersatu (Humphreys dan Lindberg 1993). Ontologi tujuan umum luas juga muncul. Misalnya, Program Pembangunan PBB dan Dun & Bradstreet menggabungkan upaya mereka untuk mengembangkan ontologi UNSPSC yang menyediakan terminologi untuk produk dan

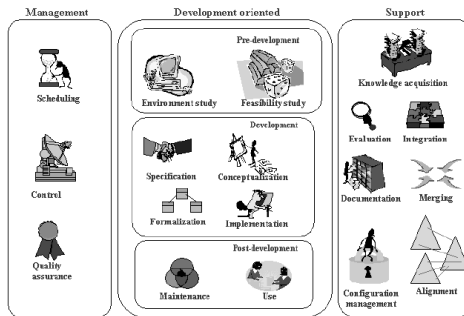
---

<sup>4</sup> N. Noy and D. McGuinness, "Ontology Development 101: A Guide to Create Your First Ontology", diakses dari [https://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html), pada tanggal 30 Desember 2018.

layanan. Terdapat beberapa manfaat dalam menggunakan ontologi, yaitu :

- a. Menjelaskan suatu domain pengetahuan secara eksplisit; memberikan struktur hierarki dari konsep untuk menjelaskan sebuah domain dan bagaimana mereka berhubungan.
- b. Berbagi pemahaman dari informasi yang terstruktur; sebagai contoh beberapa web yang berbeda memiliki informasi medis. Jika web tersebut dipakai bersama dan dipublikasikan dengan dasar ontologi yang sama maka perangkat lunak dapat mengekstrak dan mengumpulkan informasi dari situs yang berbeda.
- c. Penggunaan ulang domain pengetahuan; apabila ingin membangun ontologi yang luas dapat mengembangkan ontologi yang telah ada sebelumnya dan mengintegrasikan dengan beberapa ontologi lainnya yang relevan dengan ontologi yang ingin dibangun [3].

Pada tahun 1997, Fernández-López dan kolega nya, mengidentifikasi sebuah siklus pembangunan ontologi (*The ontology development process*). Apa yang mereka identifikasi ini berdasarkan pada standar yang dikeluarkan oleh IEEE pada tahun 1996 mengenai perancangan perangkat lunak [4]. Siklus pembangunan ontologi ini adalah aktivitas yang harus dilakukan ketika membangun sebuah ontologi. Berikut adalah diagram dari siklus pembangunan ontologi (diadaptasi dari Fernández-López dan kolega, 1997) pada gambar 1 dibawah ini :



**Gambar 2.1 Diagram Siklus Perancangan Ontologi**

## 2.2 Pemodelan

Pemodelan Data dalam rekayasa perangkat lunak adalah proses menciptakan sebuah model data dengan menerapkan model deskripsi formal data menggunakan teknik pemodelan data. Pemodelan data adalah metode yang digunakan untuk menentukan dan menganalisis persyaratan data yang diperlukan untuk mendukung proses bisnis suatu organisasi. pemodelan data mendefinisikan elemen tidak hanya data, tapi struktur dan hubungan antara mereka teknik pemodelan data dan metodologi yang digunakan untuk model data dengan cara yang standar yang konsisten, dapat diprediksi untuk mengelolanya sebagai sumber daya.

Pemodelan data merupakan sebuah tahapan dalam merancang sebuah sistem informasi. Pemodelan data berfokus pada data apa yang akan disimpan yang menggambarkan hubungan antara entiti set yang dibutuhkan oleh sebuah organisasi dalam pengelolaan data [5].

## 2.3 Protégé

Protégé adalah platform gratis yang menyediakan alat untuk membangun model domain dan aplikasi berbasis pengetahuan dengan ontologi. Protege mengimplementasikan seperangkat struktur pemodelan pengetahuan dan tindakan yang mendukung penciptaan, visualisasi, dan manipulasi ontologi dalam



berbagai format representasi. protege dapat dikustomisasi untuk memberikan dukungan untuk membuat model pengetahuan dan memasukkan data. Lebih lanjut, protege dapat diperluas melalui penambahan plug-in dan Application Programming Interface (API) berbasis Java untuk membangun aplikasi berbasis pengetahuan. Protege mendukung dua cara utama dalam memodelkan ontologi:

**Protégé frames** - memungkinkan pengguna untuk membangun dan mengisi ontologi yang berbasis bingkai, sesuai dengan protokol Open Knowledge Base Connectivity (OKBC). Dalam model ini, ontologi terdiri dari sekumpulan kelas yang diorganisasikan dalam hierarki subsumsi untuk mewakili konsep-konsep penting domain, sekumpulan slot yang terkait dengan kelas untuk menggambarkan properti dan hubungan mereka, dan sekumpulan *instance* dari kelas-kelas tersebut - contoh individual dari konsep yang memiliki nilai spesifik untuk propertinya.

**Protégé-OWL** - memungkinkan pengguna untuk membangun ontologi yang diperuntukkan untuk Web Semantik, khususnya di Web Ontology Language (OWL) W3C. "Ontologi dalam format OWL dapat mencakup deskripsi kelas, properti, dan individu nya [6].

## 2.4 OWL

Evolusi yang terjadi dengan cepat pada bahasa ontologi web semantik dimungkinkan dengan belajar dari pengalaman di mengembangkan formalisme representasi pengetahuan dan model konseptual basis data, dan dengan mewarisi dan memperluas beberapa fitur nya. Secara khusus, web semantik secara signifikan meningkatkan visibilitas dan aspek diperpanjang dari berbagai pengetahuan dibandingkan dengan pendekatan sebelumnya. Kosa kata berbasis URI dan tata bahasa berbasis XML adalah kunci utama bagi manajemen dan berbagai pengetahuan dalam skala web [7].

Salah satu hasil kuat dari web semantik pada ontologi adalah normalisasi ekspresi. Ini Intinya, penting jika kita ingin agar ontologi dapat digunakan oleh banyak orang. Meskipun sudah dikenali sebagai bahasa ontologi, kemampuan RDF agak terbatas: mereka tidak memiliki kemampuan untuk menggambarkan batasan kardinalitas, fitur yang ditemukan di sebagian besar bahasa pemodelan konseptual, atau untuk menggambarkan gabungan sederhana dari kelas. Kebutuhan akan bahasa ontologi yang lebih ekspresif diakui secara luas. Seiring waktu dilahirkan lah beberapa proposal untuk "bahasa ontologi web", contohnya SHOE, OIL dan DAML + MINYAK. World Wide Web Consortium (W3C) lalu membentuk kelompok kerja standardisasi untuk mengembangkan standar untuk Bahasa ontologi. Hasil dari kegiatan ini adalah standar bahasa ontologi OWL. Saat ini terdapat 3 tipe dari OWL :

1. OWL Lite, adalah versi OWL paling sederhana dan menyediakan hierarki klasifikasi dan batasan sederhana; hanya mengizinkan ekspresi. Untuk kardinalitas hanya dapat memiliki nilai 1 dan 0 saja. Kekurangan dari Bahasa ini adalah ekspresif nya yang terbatas. OWL Lite adalah subbahasa dari OWL DL.
2. OWL DL, menggunakan apa yang disebut deskripsi logika (Description Logic/DL), untuk mewakili hubungan antara objek dengan properti nya. Memang, OWL DL menyediakan ekspresi secara maksimum sambil menjaga kelengkapan sifat.
3. OWL Full, Subbahasa OWL Full memberikan ekspresi tertinggi dan kebebasan sintaksis dari RDF. OWL Lite dan OWL DL adalah subbahasa dari OWL Full.

Bisa dibilang bahwa OWL adalah Bahasa Ontologi yang sangat lengkap. Karena didalamnya, terdapat karakteristik berikut :

1. Konsep, dapat mendefinisikan :
  - a. Partisi, serta kelas
  - b. Atribut, meskipun tidak mungkin untuk dibedakan antara atribut class dan *instance*. Mereka bisa didefinisikan baik untuk lingkup lokal maupun global dengan menentukan domain mereka.
  - c. Facets, atau atribut properti, seperti tipe dari Constraint atau kardinalitas, serta dokumentasi.
  
2. Taksonomi, dapat mendefinisikan sub kelas, dengan banyak pewarisan secara lengkap, Pada saat yang sama, sebuah kelas bisa didefinisikan sebagai pelengkap dari yang lain.
  
3. Aksioma, OWL memungkinkan definisi logika aksioma sehubungan dengan hubungan properti aljabar (simetri, transitivitas, dan keunikan). Keuniversalan dan eksistensialitas juga dapat didefinisikan untuk kelas dan properti.

Kemudian, bahasa ontologi ini dapat langsung digunakan untuk menentukan manajemen informasi, karena memiliki sebagian besar konstruksi:

1. Kelas dapat didefinisikan menggunakan tag owl: Class, dan Sub kelas nya dengan rdfs: subclassOf
2. atribut ditentukan menggunakan owl: DatatypeProperty untuk nilai literal atau owl: ObjectProperty sebagai referensi ke kelas.. Salah satu ciri khas dari OWL adalah bahwa properti tidak secara langsung milik kelas. Kemudian, untuk melampirkan properti ke sebuah kelas, dapat ditambahkan tag rdfs: domain [8].

## 2.5 SPARQL

SPARQL adalah Bahasa query untuk graf. Bila ada data D, sebuah query yang terdiri dari pola yang cocok dengan D, maka data yang diperoleh dari pencocokan ini akan di proses sebagai jawaban. Sumber data dari D yang akan di query dapat terdiri dari dari berbagai sumber yang berbeda [9]. SPARQL terdiri dari beberapa bagian :

1. Prefix Declaration, Untuk memasukkan dan menyingkat URI
2. Dataset Definition, menyatakan graf RDF mana yang akan di query kan
3. Result Clause, mengidentifikasi informasi apa yang akan di return dari query
4. Query Pattern, menspesifikan apa yang akan di query kan dari dataset
5. Query Modifiers, untuk mengurutkan dan merapihkan hasil query<sup>5</sup>.

Terdapat beberapa tipe sintaks pada SPARQL :

1. Read Oriented Query
  - a. Select, mirip seperti select pada SQL, yang menghasilkan solusi dari query dengan format tabel
  - b. Construct, mengambil bentuk kalimat dari RDF
  - c. Ask, tipe ini mengembalikan nilai dari Boolean (ya dan tidak)
2. Write Oriented Query
  - a. Create, membuat graf baru

---

<sup>5</sup> L. Feigenbaum, "SPARQL by Example", diakses dari <https://www.w3.org/2009/Talks/0615-qbe/>, pada tanggal 16 Juni 2019.

- b. Insert, menambahkan kalimat RDF kepada graf yang sudah dinamai berdasarkan suatu kondisi
- c. Copy, mengkopi kalimat RDF antara 2 graf
- d. Add, menambahkan RDF ke dalam graf
- e. Move, memindahkan kalimat RDF dari suatu graf yang sudah dinamai ke graf lain yang sudah dinamai. Graf asal dan kalimat RDF nya akan dihapus secara permanen apabila operasi ini berhasil
- f. Delete, menghapus kalimat RDF. Dapat mentargetkan pernyataan yang spesifik
- g. Clear, Menghapus semua RDF dalam sebuah graf. Hasil akhirnya adalah graf yang memiliki nama namun kosong
- h. Drop, menghapus graf dari triple store<sup>6</sup>.

Berikut adalah contoh penggalan sintaks dari SPARQL :

```
# prefix declarations
PREFIX foo: <http://example.com/resources/>
...
# dataset definition
FROM ...

# result clause
SELECT ...

# query pattern
WHERE {
  ...
}

# query modifiers
ORDER BY ...
```

<sup>6</sup> K. U. Idehen, "What is a SPARQL Endpoint, and why is it important?", diakses dari <https://medium.com/virtuoso-blog/what-is-a-sparql-endpoint-and-why-is-it-important-b3c9e6a20a8b>, pada tanggal 7 Juni 2019.

### **Kode Sumber 2.1 Contoh Penggalan Sintaks dari SPARQL**

Hasil akhir dalam SPARQL query bisa di kembalikan dalam beberapa format yang berbeda seperti: XML, JSON, RDF, dan HTML. Sebuah SPARQL endpoint menerima query dan mengembalikan hasilnya via HTTP<sup>7</sup>.

## **2.6 RDF**

RDF adalah bahasa yang digunakan untuk memberikan standar metadata tentang sumber daya di Web, yang mampu mewakili data dan bertukar pengetahuan melalui Web. RDF dikembangkan agar dapat dipahami dengan mudah oleh komputer, dan memfasilitasi interoperabilitas antara aplikasi. Dengan kata lain, ini adalah kerangka kerja untuk menggunakan dan mewakili metadata dan menggambarkan hal semantik mengenai informasi tentang sumber daya web dengan cara yang dapat dimengerti oleh mesin.

RDF, yang direkomendasikan oleh W3C, menggunakan URI untuk mengidentifikasi sumber daya atau benda (*Root* dari sebuah ontologi disebut dengan *Thing*). URI dirancang berdasarkan XML, yang diperuntukkan untuk sintaks. Sedangkan RDF lebih kepada hal semantik. Seperti yang telah disebutkan, RDF adalah kerangka kerja untuk menggambarkan sumber daya web, itulah sebabnya RDF telah menjadi metode umum untuk menggambarkan properti, waktu, informasi, dan konten sumber daya web, sehingga dapat dibaca dan dipahami oleh aplikasi computer.

RDF dapat digunakan dalam beberapa aplikasi, salah satu yang paling penting adalah dapat mencari sumber daya

---

<sup>7</sup> L. Feigenbaum, "SPARQL by Example", diakses dari <https://www.w3.org/2009/Talks/0615-qbe/>, pada tanggal 16 Juni 2019.

pengetahuan. Ini dapat digunakan untuk meningkatkan kemampuan dari mesin pencari. RDF juga digunakan untuk memfasilitasi berbagi pengetahuan dan pertukaran perangkat lunak cerdas, dan seperti yang disebutkan sebelumnya, untuk menggambarkan konten dan hubungan konten yang tersedia dengan sumber daya apa pun, seperti halaman (*page*).

RDF memiliki tiga elemen: subjek, objek dan predikat. Dimungkinkan bahwa *<subject>* memiliki property *<predicate>* dilanjutkan oleh *<object>*. Contoh sintaks dari RDF adalah sebagai berikut [10] :

```
<RDF>
  <Description about="https://www.w3schools.co
m/rdf">
    <author>Jan      Egil      Refsnes</author>
    <homepage>https://www.w3schools.com</homep
age>
  </Description>
</RDF>
```

#### Kode Sumber 2.2 Contoh Sintaks dari RDF

1. Resource, adalah sesuatu yang dapat memiliki URI, pada contoh diatas, <https://www.w3schools.com/rdf> adalah subjek
2. Property adalah Sumber yang memiliki nama, seperti "penulis" atau "beranda"
3. Property Value, seperti "Jan Egil Refsnes" atau <https://www.w3schools.com>

## 2.7 EasyRDF

EasyRDF adalah *Library* PHP yang digunakan agar RDF dengan mudah diproduksi dan mengonsumsi data yang telah disimpan. EasyRDF adalah sebuah *Library* algoritma yang ditulis dalam bahasa PHP berbasis *Object Oriented Programming*. *Library* ini memudahkan kinerja query SPARQL yang ditulis melalui HTTP menjadi *triplestore* menggunakan *class* EasyRdf Sparql Client untuk mengembalikan objek PHP sebagai hasil.

Terdapat bermacam-macam *method* yang dapat dipanggil dari *Library* EasyRDF untuk menuliskan *query* sesuai fungsi yang diinginkan. Berdasarkan dokumentasi dari *website* EasyRDF<sup>8</sup> Tabel berikut akan menjelaskan beberapa *method* yang digunakan pada pengerjaan tugas akhir ini beserta fungsi yang dimilikinya.

Method	Fungsi
EasyRdf_Graph()	Referensi API yang mendukung penguraian data RDF menjadi <i>graph</i>
EasyRdf_Namespace::setDefault()	Mendeskripsikan namespace yang digunakan
parseFile()	Menguraikan suatu data RDF menjadi objek <i>graph</i>
get()	Mendapatkan sebuah nilai dari <i>property</i> suatu <i>resource</i>
all()	Mendapatkan semua nilai dari <i>property</i>
allOfType()	Mendapatkan semua <i>Resource</i> dalam suatu <i>graph</i> dengan tipe atau <i>class</i> tertentu
Resource()	Mendapatkan atau membuat suatu <i>resource</i> yang tersimpan dalam <i>graph</i>

---

<sup>8</sup> EasyRDF, "EasyRdf API Documentation", diakses dari [http://www.easyrdf.org/docs/api/EasyRdf\\_Graph.html](http://www.easyrdf.org/docs/api/EasyRdf_Graph.html), pada tanggal 14 Juni 2019.



Berikut merupakan contoh penggunaan *Library EasyRDF* untuk mendapatkan semua *Resource* dengan tipe 'Foody' dari *namespace* foo yang sudah menjadi *graph*.

```
$graph->allOfType('foo:Foody')
```

Tanda ":" digunakan sebagai penghubung antara *namespace* dan *Resource* yang berkaitan. Sedangkan, sebuah nilai yang didapatkan ditandai dengan ekspresi " ' ' ".

## 2.8 RDFlib

RDFlib adalah *Library* Python yang khusus untuk menghandle RDF. RDFlib berisi sebagian besar hal yang dibutuhkan untuk bekerja dengan RDF, diantara nya adalah<sup>9</sup> :

1. *Parser* dan *serializers* untuk RDF / XML, N3, NTriples, Quads, Turtle, TriX, RDFa dan Microdata.
2. antarmuka grafik yang dapat didukung oleh salah satu dari sejumlah store implementation.
3. menyimpan implementasi untuk penyimpanan memori dan penyimpanan persisten di atas Berkeley DB.
4. Mendukung bekerja dengan SPARQL

RDFlib diciptakan oleh Daniel Krech, 16 tahun yang lalu. Saat ini versi stabil nya adalah versi 4.2.2 yang dirilis 2 tahun yang lalu. RDFlib tergolong simpel, namun cukup powerful untuk mewakili informasi. Selain yang sudah disebutkan diatas, *Library* ini juga mendukung *Both in-memory* dan *persistent Graph backends*.

## 2.9 PHP

---

<sup>9</sup> RDFlib, "RDFlib Documentation", diakses dari //rdflib.readthedocs.io/en/stable/, pada tanggal 15 June 2019.

PHP (PHP: *Hypertext Preprocessor*) adalah *open source* bahasa *scripting* yang mempunyai tujuan umum secara khusus untuk pengembangan web<sup>10</sup>. PHP juga dapat ditanamkan ke dalam HTML. PHP tidak seperti bahasa C untuk pengembangan *web*. Namun struktur sintaks dasarnya sama, sehingga fleksibel dan mudah untuk diimplementasikan [11].

PHP adalah bahasa pemrograman dengan berbagai fitur yang komprehensif dan juga mendukung orientasi objek. Saat ini, PHP lebih sering disebut sebagai bahasa pemrograman dinamis. Berbeda bahasa pemrograman lainnya yang bersifat tradisional seperti C / C++, kompilasi tidak diperlukan oleh PHP. Salah satu kelebihan yang dimiliki oleh PHP adalah kebebasan *platform*. PHP *compatible* untuk berbagai sistem operasi seperti Linux (untuk berbagai arsitektur CPU), Microsoft Windows, Mac OS X, Sun Solaris (SPARC dan Intel), IBM AIX, UX-HP, FreeBSD, Novell Netware, SGI, IRIX, IBM AS/400, OS/2 dan RISC OS. Keuntungan lain yang diberikan PHP adalah fleksibilitas. *Bug* yang terjadi dapat dengan mudah dirubah atau diperbaiki dalam beberapa menit karena tidak diperlukan proses kompilasi dan mampu menciptakan versi baru dari program secara bertahap [12].

Untuk berhubungan dengan *web browser*, PHP memiliki *method* yang dapat dipanggil sesuai fungsinya. Terdapat dua cara yang digunakan oleh *browser client* untuk mengirimkan informasi pada *web server*, yaitu *GET method* dan *POST method*. Sebelum browser mengirim informasi, *browser* mengkonversi informasi tersebut menggunakan sebuah skema yang disebut *URL encoding*. *GET method* mengirimkan informasi pengguna yang telah dikodekan untuk ditambahkan pada *page request*. Halaman dan kode informasi dipisah oleh karakter "?". Sedangkan *POST method* memindahkan informasi melalui header HTTP.

---

<sup>10</sup> Achour, F. Betz, A. Dovgal, N. Lopes, H. Magnusson, G. Richte, D. S. and J. Vrana, "PHP Manual", diakses dari <http://php.net/manual/en/>, pada tanggal 16 Juni 2019.

Kode informasi yang diberikan oleh *GET method* kemudian dimasukkan ke dalam *header* yang disebut *QUERY\_STRING*. Selain itu, pengguna juga dapat menyertakan isi dari sebuah berkas PHP ke dalam berkas PHP lain sebelum *server* mengeksekusinya. Salah satunya adalah fungsi `require()` yang mengambil semua teks dalam *file* tertentu dan menyalinnya dalam *file* yang menggunakan fungsi `include()`<sup>11</sup>.

## 2.10 Python

Python adalah bahasa pemrograman interpreter tingkat tinggi, berorientasi objek, dengan tingkat semantik yang dinamis. Struktur data nya yang dibangun dengan tingkat tinggi, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk Pengembangan Aplikasi Cepat, serta untuk digunakan sebagai bahasa scripting atau perekat untuk menghubungkan komponen yang ada bersama-sama. Sintaks Python yang sederhana dan mudah dipelajari menekankan pembacaan dan karenanya mengurangi biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode program. Interpreter Python dan *Library* yang luas tersedia dalam bentuk source atau biner, gratis untuk semua platform, dan dapat didistribusikan secara bebas<sup>12</sup>.

Dibuat oleh Guido Van Rossum pada tahun 1991 python dapat digunakan untuk mengembangkan web, pengembangan aplikasi, memecahkan problem matematika, dan scripting. Versi yang saat ini paling populer adalah versi 3. Namun, masih banyak juga yang memakai versi 2.

---

<sup>11</sup> T. Point, "PHP Hypertext Preprocessor", diakses dari [www.tutorialspoint.com/php/php\\_tutorial.pdf](http://www.tutorialspoint.com/php/php_tutorial.pdf), pada tanggal 10 Juni 2019.

<sup>12</sup> Python, "What is Python? Executive Summary", diakses dari <https://www.python.org/doc/essays/blurb/>, pada tanggal 18 Juni 2019.

Beberapa pertimbangan yang digunakan dalam memilih python sebagai Bahasa pemrograman adalah sebagai berikut<sup>13</sup>:

1. Dapat berjalan pada banyak platform.
2. Memiliki sintaks yang simpel seperti Bahasa inggris.
3. Dikarenakan sintaks nya yang simpel, memungkinkan programmer untuk menulis baris kode yang lebih pendek dibanding dengan menggunakan Bahasa pemrograman lainnya.
4. Python berjalan pada sistem interpreter, yang berarti dapat dijalankan langsung setelah kode ditulis. Menjadikan nya sangat cepat untuk mengembangkan prorotype.
5. Kode program dapat ditulis dengan cara procedural, berbasis object, atau cara fungsional.

Python, saat ini adalah keterampilan yang paling banyak diminta dan juga keterampilan dengan gaji rata-rata tertinggi yang ditawarkan di industri. Dengan munculnya *Big Data*, *Programmer* Python saat ini banyak dilirik sebagai *Data Scientist*, terutama karena Python dapat dengan mudah diintegrasikan ke dalam aplikasi web untuk melakukan tugas yang membutuhkan *Machine Learning* [13].

---

<sup>13</sup> w3school, "Python Introduction", diakses dari [https://www.w3schools.com/python/python\\_intro.asp](https://www.w3schools.com/python/python_intro.asp), pada tanggal 20 June 2019.

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Pada bab 3 ini akan dijelaskan mengenai analisis dan perancangan ontologi beserta sistem untuk mencapai tujuan dari Tugas Akhir. Perancangan ini meliputi analisis metode secara umum, analisis dan perancangan ontologi, serta analisis dan perancangan sistem.

#### **3.1 Analisis Dataset**

Pengetahuan yang akan dimodelkan pada ontologi diperoleh dari sumber-sumber diluar, seperti dari google, website resep makanan seperti cookpad, dan yang paling besar porsi nya, adalah data yang diberikan oleh gifood. Mengenai data dari gifood, dari pihak gifood memberikan data pendukung untuk membangun ontologi ini berupa data dengan format .csv yang bernama *foods.csv*. Isi dari data ini adalah transaksi makanan yang terjadi pada beberapa bulan terakhir sebelum pengerjaan tugas akhir ini dimulai. Data tersebut mengandung beberapa informasi yang cukup penting seperti kolom judul makanan dan deskripsi dari makanan. Perancangan ontologi kali ini, fokus kepada 2 kolom tersebut, dikarenakan kedua kolom tersebut mengandung informasi yang sangat penting, dan tidak bisa hanya memakai salah satu dari kolom tersebut.

Isi kolom dari judul makanan memberikan gambaran awal mengenai makanan apa yang dibagikan, namun biasanya data tersebut kurang jelas, seperti contoh Nasi kotak dan Kue. Dari kedua data tersebut, kita bisa mengetahui apa makanan yang dibagikan, namun tetap saja kurang jelas apa isi dari makanan tersebut. Maka di kolom kedua, deskripsi makanan, kita bisa mengetahui dengan jelas apa saja yang terkandung di dalam makanan yang disebut. Sebagai contoh, untuk Nasi Kotak, pada deskripsi ternyata dijelaskan bahwa isi nya adalah : Nasi Ayam Goreng, dan sayur sosis. Maka kita bisa mendapat informasi bahwa

ada 3 jenis makanan : Nasi, Ayam Goreng, dan Sayur Sosis dan 1 jenis kemasan makanan : Kotak Makanan. Untuk contoh kedua yaitu Kue, pada deskripsi juga dijelaskan bahwa kue tersebut adalah kue coklat.

Ekstraksi data dari *foods.csv* dilakukan dengan manual menggunakan tools pembantu Microsoft office Excel. Fokus dengan hanya kepada 2 kolom Judul Makanan dan Deskripsi. Penulis disini secara manual, mengambil kata-kata benda yang berhubungan dengan makanan lalu membuat list dari makanan yang ada di dalam transaksi. Berdasarkan list makanan yang ada, lalu dirancang juga kelas, *object properties*, *data properties*, dan *instance* pada ontologi yang akan dibangun

### 3.2 Analisis dan Perancangan Ontologi

**Tabel 3.1 Literature Review Penelitian Terdahulu**

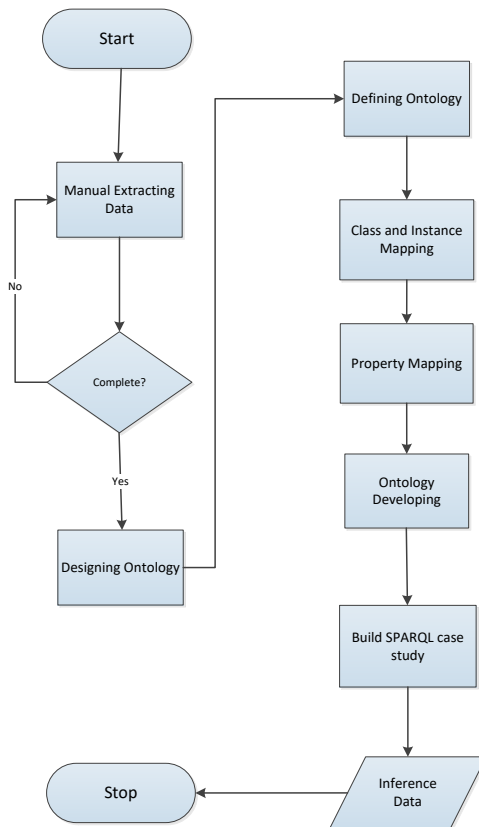
Judul Paper atau penelitian	Deskripsi garis besar	kelebihan	Kekurangan
Overview Of Methodologies For Building Ontologies	Menganalisis beberapa metodologi dalam merancang ontologi	Memberikan gambaran awal yang cukup jelas mengenai beberapa metodologi dalam merancang Ontologi	Hanya sekedar membandingkan dan tidak memberikan penjelasan yang lebih mendalam
FOODS: A Food-Oriented Ontology-Driven System	Desain dan pengembangan dari sistem konseling untuk perancangan menu makanan dengan	Mencantumkan dan menggambarkan ontologi yang dirancang dengan jelas	Kurangnya informasi mengenai informasi nutrisi pada tiap

	menggunakan ontologi	beserta skemanya.	kandungan makanan
User's Profile Ontology-Based Semantic Framework for Personalized Food and Nutrition Recommendation	Merancang sebuah sistem untuk merekomendasikan Makanan menggunakan Ontologi	Hasil dari implementasi menunjukkan bahwa sistem dapat memberikan rekomendasi dengan akurat	Sistem masih harus ada yang diperbaiki agar bisa dilepas ke publik

Pada Tugas Akhir ini, Ontologi akan dibangun dari nol. Pertimbangan utama untuk membangun ontologi ini dari nol adalah, ontologi yang dibangun menyesuaikan dengan studi kasus dari aplikasi gifood. Karena pada tugas akhir ini pada dasarnya merancang *Food Ontology*, dan dikarenakan ranah dari aplikasi gifood adalah pada wilayah DI Yogyakarta, maka pengetahuan mengenai makanan yang dimodelkan harus juga menyesuaikan dengan ranah dari aplikasi gifood. Tidak mungkin untuk memodelkan makanan Jepang, India, atau Italia karena berbeda dengan ranah dari aplikasi gifood. Dan berdasarkan dari pencarian penulis, tidak menemukan satu pun ontologi yang terkait dengan studi kasus kali ini, yang mungkin bisa dijadikan patokan atau referensi. Pada tabel 3.1, dijelaskan *review* dari literatur penelitian terdahulu. Terdapat beberapa literatur yang dijelaskan. Namun terdapat satu jurnal yang cukup menarik untuk dijadikan acuan untuk mendesain ontologi untuk tugas akhir ini. Jurnal berjudul *FOODS: A Food-Oriented Ontology-Driven System* yang ditulis oleh Chakkrit Snae dan Michael Brückner [2] ini salah satunya membahas apa saja yang sebaiknya dimodelkan dalam ontologi bila kasusnya adalah memodelkan seputar dunia kuliner. Hal ini dapat dijadikan salah satu referensi mengenai apa saja yang

seharusnya dimodelkan ke dalam ontologi dan sebagai apa bentuknya, apakah sebagai kelas, relasi, atribut, atau individu.

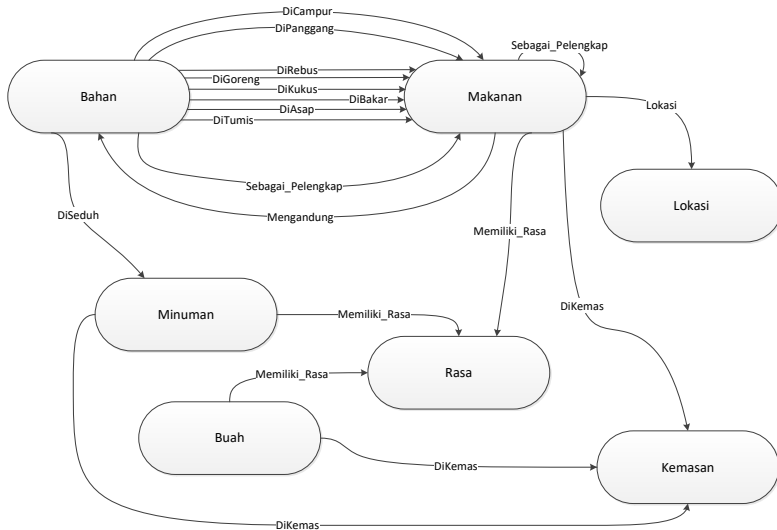
Ontologi pada tugas akhir ini dibangun dengan menggunakan tools pembantu Protégé. Pada gambar 3.1 dibawah ini dijelaskan alur pada perancangan ontologi yang dilakukan pada tugas akhir ini. Adapun penjelasan dari gambar 3.1 akan dijabarkan lebih detail pada subbab berikutnya.



**Gambar 3.1 Alur Perancangan Ontologi**



Gambar 3.2 akan menunjukkan bagaimana hubungan antara kelas dan property objek pada rancangan ontologi yang akan dibangun pada tugas akhir ini.



**Gambar 3.2 Diagram rancangan ontologi**

Gambar 3.2 menunjukkan hubungan antara kelas yang ada pada ontologi, beserta relasi (*Object Properties*) apa saja yang menghubungkan kelas tersebut. Dapat dilihat bahwa relasi paling banyak, adalah relasi yang menghubungkan antara kelas Makanan dan Bahan. Hal tersebut dapat terjadi karena kedua kelas tersebut memegang peranan yang sangat penting pada ontologi ini. Kedua kelas tersebut bisa dikatakan sebagai “inti” dari ontologi ini.

Pada subbab 3.1, sudah dijelaskan bahwa pengambilan data dari dataset *foods.csv*, hanya menggunakan 2 kolom yang ada, yaitu nama transaksi dan deskripsi dari transaksi tersebut untuk kemudian diambil kata benda yang relevan dengan studi kasus. Sebagian besar dari kata benda tersebut, akan masuk ke kelas Makanan, Minuman, dan Buah. Walaupun ada sebagian kecil juga

yang masuk ke kelas Kemasan, dan tidak terlalu mempengaruhi hasil akhir dari analisis, namun kelas Makanan menjadi kelas terbanyak yang menampung kata benda dari dataset. Hal ini dapat terjadi karena sebagian besar transaksi yang terjadi pada gifood melibatkan makanan, bukan minuman, buah, ataupun kemasan.

Selanjutnya, setelah kelas makanan, minuman, dan buah sudah terbentuk. Maka selanjutnya adalah bagaimana bisa untuk memodelkan informasi yang lebih banyak, dengan tentunya melibatkan ketiga kelas ini. Solusi nya adalah dengan menciptakan kelas Bahan, yang nantinya akan berelasi dengan kelas makanan, dan minuman, sehingga ada informasi baru yang nantinya berguna. Untuk kelas bahan, isinya adalah penguraian lebih detail dari instance pada kelas makanan dan minuman. Sumber data untuk kelas bahan berasal dari website untuk dunia kuliner seperti cookpad, ResepKoki, dan selerarasa. Sebagai contoh, untuk instance gudeg, terdapat bahan yang biasanya dipakai sepertiangka muda, kecap manis, bawang putih, dan bawang merah. Dan gudeg, cara memasaknya adalah mengukus bahan-bahan tersebut. Maka selanjutnya, bahan-bahan tadi menjadi instance pada kelas Bahan, dan bahan-bahan tersebut, memiliki relasi “DiKukus” kepada instance “Gudeg” yang berada di kelas Makanan.

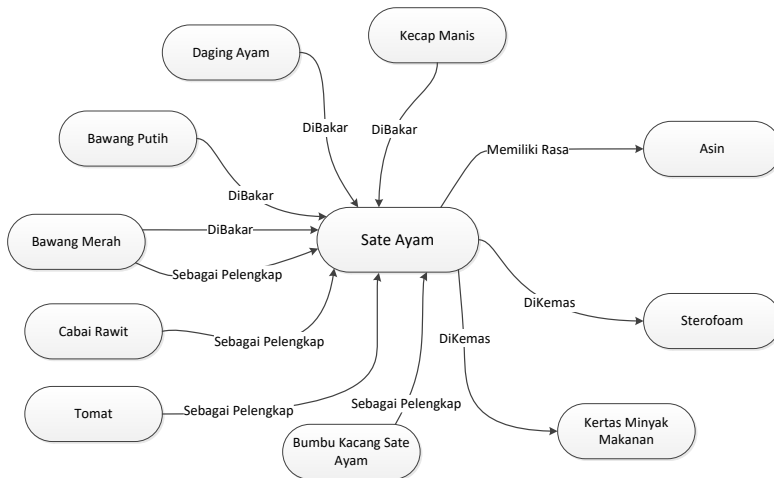
Untuk kelas Lokasi, tentu nya hal ini sebagai efek dari studi kasus dari Tugas Akhir ini. Penulis membayangkan bahwa suatu saat nanti, ontologi dan sistem yang dibangun pada Tugas Akhir ini, akan diintegrasikan dengan sistem yang dimiliki oleh gifood. Maka dari itu, diharuskan ada “jalan masuk” kepada ontologi yang dikembangkan, untuk nanti nya dapat berintegrasi dengan sistem gifood. Harus ada kesamaan data yang nantinya dapat menjadi “jalan masuk”. Satu data yang dapat menjadi “jalan masuk” tersebut adalah data mengenai lokasi. Data lokasi ini adalah data yang menerangkan lokasi dari transaksi dari suatu makanan pada sistem gifood. Di dalam ontologi, kelas lokasi berelasi dengan kelas Makanan.

Untuk kelas Kemasan, penulis melihat dari kasus transaksi yang terjadi. Semua transaksi yang terjadi pada gifood, yang

melibatkan Makanan, Minuman, dan Buah, dipastikan memakai kemasan untuk kemudian Makanan, Minuman, dan Buah tersebut diberikan kepada yang membutuhkan. dengan ini dapat disimpulkan bahwa informasi kemasan juga dibutuhkan pada studi kasus ini. Terdapat juga pengaruh kemasan yang dipakai terhadap ketahanan dari suatu makanan. Hal ini dapat berpengaruh terhadap seberapa makanan tersebut dapat bertahan dan tidak menjadi basi [14]. Pada ontologi yang dibangun, kelas Kemasan akan memiliki relasi dengan kelas Makanan, Minuman, dan Buah.

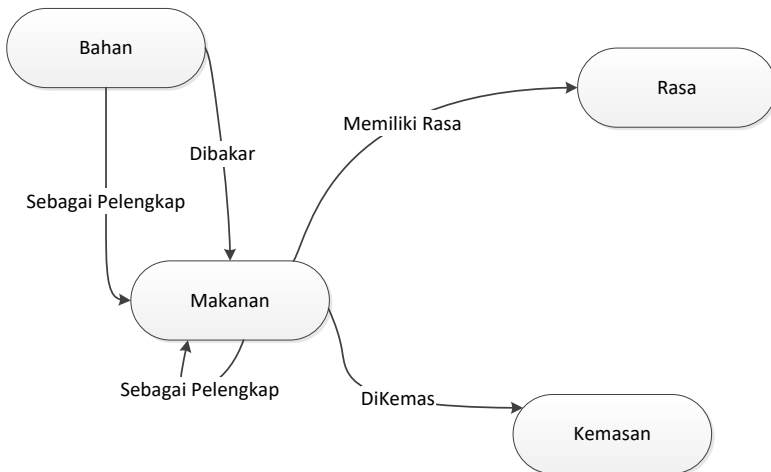
Selanjutnya untuk kelas Rasa, menurut pengalaman pribadi penulis, informasi penting yang biasanya orang-orang simpan dari suatu makanan, adalah rasa dari makanan tersebut. Biasanya ketika orang menanyakan mengenai suatu makanan, hal yang biasanya ditanya adalah “bagaimana Rasa nya?” atau “Apakah Rasa nya Enak?”, atau “Rasanya Manis kah? Atau asin?”. Pertanyaan diatas sudah pasti berkaitan dengan rasa dari suatu makanan. Hal ini juga dikuatkan dari penelitian yang menyebutkan bahwa faktor Rasa memiliki pengaruh yang cukup penting dalam pertimbangan manusia untuk mengkonsumsi suatu makanan [15]. Maka dapat disimpulkan bahwa Rasa adalah suatu informasi yang sangat penting dalam dunia kuliner. Hal ini yang mendorong untuk diimplementasikannya kelas rasa. Pada ontologi yang dibangun, kelas Rasa akan memiliki relasi dengan kelas Makanan, Minuman, dan Buah.

Agar dapat melihat hubungan yang jelas untuk 1 makanan, maka penulis mengambil contoh untuk Makanan Sate Ayam. Gambar 3.3 akan menunjukkan contoh hubungan untuk Makanan Sate ayam. dan gambar 3.4 akan menunjukkan hubungan kelas yang terlibat dari makanan Sate Ayam.



**Gambar 3.3 Contoh Hubungan untuk Instance “Sate Ayam”**

Pada Gambar 3.3 menunjukkan bahwa untuk instance “Sate Ayam”, memiliki hubungan ke instance lain diluar dari kelas Makanan. Disini dapat diambil informasi bahwa sate ayam memiliki kurang lebih 8 bahan dimana 4 diantaranya bersifat sebagai pelengkap. Yang dimaksud disini sebagai pelengkap adalah, bahwa instance tersebut termasuk bagian dari makanan sate ayam, namun pengolahannya terpisah atau tidak ikut “DiBakar” seperti halnya sate ayam, entah itu dapat dimakan mentah (bawang merah, cabai rawit, tomat), atau memiliki cara penyajian yang berbeda (bumbu kacang sate ayam). Sate ayam, menurut beberapa sumber di internet, memiliki rasa yang gurih dan asin, dan biasanya dikemas dengan sterofoam, atau kertas minyak untuk makanan. Untuk memodelkan sate ayam seperti pada gambar 3.3, melibatkan cukup banyak kelas, maka pada gambar 3.4 akan ditunjukkan mengenai kelas apa saja yang terlibat untuk contoh “Sate Ayam”



**Gambar 3.4 Instance yang Terlibat pada Contoh “Sate Ayam”**

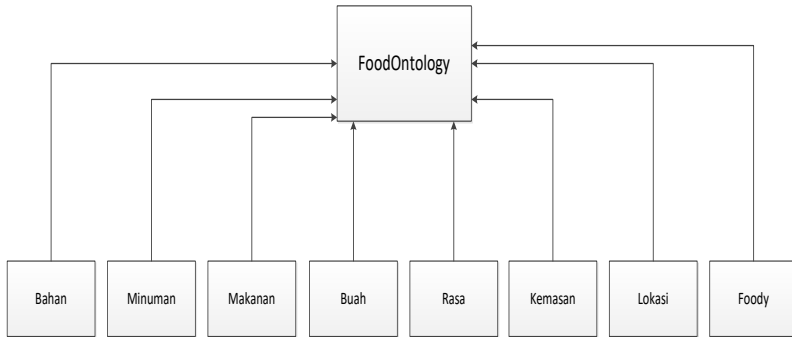
Pada gambar 3.4 ditunjukkan lebih rinci mengenai kelas apa saja yang terlibat pada contoh “Sate Ayam”. Bila kita lihat lebih detail lagi, pada kelas Makanan, terdapat relasi “Sebagai Pelengkap” yang menunjuk ke kelas nya sendiri. Hal ini terjadi untuk instance “Bumbu Kacang Sate Ayam” yang memiliki relasi “Sebagai Pelengkap” ke instance “Sate Ayam”.

Uraian diatas adalah mengenai gambaran awal dari ontologi yang dibangun. Pada subbab 3.2.1-3.2.5 akan dijelaskan lebih lanjut mengenai isi dari ontologi. Sedangkan pada subbab 3.2.6 akan dijelaskan lebih detail mengenai query yang nantinya akan dipakai untuk uji coba dari ontologi.

### 3.2.1 Deskripsi Kelas

Pada Ontologi yang dibangun kali ini, terdapat 8 kelas yang dirancang. Kelas yang dirancang sudah menyesuaikan dengan studi kasus yang diambil pada Tugas Akhir ini. Pada subbab sebelumnya juga sudah dijelaskan mengenai pertimbangan dan analisis mengenai pemilihan dari kelas pada ontologi ini. Gambar 3.5 akan menunjukkan hierarki kelas yang ada pada

ontologi kali ini. Sedangkan list kelas yang ada pada ontologi kali ini akan diuraikan dalam tabel 3.1.



**Gambar 3.5 Hierarki Kelas pada Ontologi**

**Tabel 3.2 Deskripsi Kelas**

KELAS		
Bahan	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bahan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bahan</a>
	Fungsi	Mendeskripsikan <i>instance</i> sebagai bahan dari suatu makanan, atau minuman
Buah	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Buah">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Buah</a>
	Fungsi	Mendeskripsikan <i>instance</i> sebagai buah
Foody	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Foody">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Foody</a>
	Fungsi	Mendeskripsikan <i>instance</i> untuk kemudian ditampilkan di sistem informasi
Kemasan	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kemasan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kemasan</a>
	Fungsi	Mendeskripsikan <i>instance</i> sebagai kemasan dari suatu makanan, minuman, atau buah
Lokasi	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lokasi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lokasi</a>

	Fungsi	Mendeskripsikan <i>instance</i> sebagai lokasi dari transaksi (sebagai jalan masuk untuk integrasi dengan sistem gifood nantinya)
Makanan	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Makanan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Makanan</a>
	Fungsi	Mendeskripsikan <i>instance</i> sebagai makanan
Minuman	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Minuman">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Minuman</a>
	Fungsi	Mendeskripsikan <i>instance</i> sebagai minuman
Rasa	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Rasa">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Rasa</a>
	Fungsi	Mendeskripsikan <i>instance</i> sebagai rasa dari suatu makanan, minuman, atau buah

### 3.2.2 Deskripsi *Object Properties*

*Object properties* atau yang bisa disebut juga sebagai relasi, menghubungkan antara 1 individu/*instance* dengan individu yang lainnya. Pada ontologi yang akan dibangun, tabel 3.2 akan menguraikan *object properties* yang akan digunakan.

**Tabel 3.3 Deskripsi *Object Properties***

<i>Object Properties</i>		
DiAsap	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiAsap">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiAsap</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
DiBakar	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiBakar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiBakar</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah

	Kelas	Bahan, Makanan, Minuman
DiCampur	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiCampur">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiCampur</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
DiGoreng	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiGoreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiGoreng</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
DiKemas	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiKemas">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiKemas</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Makanan, Minuman, Buah, Kemasan
DiKukus	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiKukus">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiKukus</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
DiPanggang	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiPanggang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiPanggang</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
DiRebus	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiRebus">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiRebus</a>
	Fungsi	Mendeskripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman



DiSeduh	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiSeduh">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiSeduh</a>
	Fungsi	Mendesripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
DiTumis	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiTumis">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#DiTumis</a>
	Fungsi	Mendesripsikan cara penyajian dari suatu makanan, minuman, dan buah
	Kelas	Bahan, Makanan, Minuman
Lokasi	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lokasi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lokasi</a>
	Fungsi	Mendesripsikan lokasi dari sebuah transaksi makanan
	Kelas	Lokasi, Makanan
Memiliki_Rasa	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Memiliki_Rasa">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Memiliki_Rasa</a>
	Fungsi	Mendesripsikan rasa dari suatu makanan, minuman, dan buah
	Kelas	Makanan, Buah, Minuman, Rasa
Mengandung	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mengandung">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mengandung</a>
	Fungsi	Mendesripsikan Kandungan Bahan pada suatu Makanan dan minuman
	Kelas	Makanan, Minuman, Buah
Sebagai_Pelengkap	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sebagai_Pelengkap">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sebagai_Pelengkap</a>
	Fungsi	Mendesripsikan pelengkap dari suatu makanan
	Kelas	Bahan, Makanan

Semua object Properties pada ontologi kali ini, memiliki karakteristik yang sama, yaitu *Asymmetric* dan *Irreflexive*. Penjelasan dari kedua nya adalah sebagai berikut :

1. *Asymmetric*, bila ada individu X memiliki relasi ke individu Y, maka individu Y tidak berelasi terhadap individu X untuk property yang sama.
2. *Irreflexive*, setiap individu tidak bisa memiliki relasi terhadap diri nya sendiri melalui property yang sama.

### 3.2.3 Deskripsi *Data Properties*

*Data properties* atau atribut digunakan untuk menyimpan informasi dari sebuah individu. Tabel 3.3 akan menjelaskan lebih rinci mengenai data properties yang ada pada ontologi ini.

**Tabel 3.4 Deskripsi *Data Properties***

<i>Data Properties</i>		
Jumlah_Porsi	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jumlah_Porsi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jumlah_Porsi</a>
	Fungsi	Mendeskripsikan jumlah porsi dari suatu makanan
Kadaluarsa_Dalam_Hari	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kadaluarsa_Dalam_Hari">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kadaluarsa_Dalam_Hari</a>
	Fungsi	Mendeskripsikan berapa lama suatu makanan, minuman, dan buah aman dikonsumsi
Temperatur	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Temperatur">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Temperatur</a>
	Fungsi	Mendeskripsikan temperatur dari suatu makanan, dan minuman

### 3.2.4 Deskripsi *Instance*

Pada bagian ini akan di jelaskan lebih rinci mengenai *instance/individual* yang terdapat pada ontologi. Dikarenakan jumlah nya cukup banyak, maka pada subbab ini hanya akan ditampilkan beberapa contoh instance saja. Untuk list lengkap dari nama instance dapat dilihat pada lampiran.

**Tabel 3.5 Deskripsi Instance**

<i>Instance</i>		
Bawang_Putih	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Putih">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Putih</a>
	Kelas	Bahan
Daging_Sapi	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Sapi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Sapi</a>
	Kelas	Bahan
Jeruk	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk</a>
	Kelas	Buah
Salak	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Salak">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Salak</a>
	Kelas	Buah
Kertas_Minyak_Makanan	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kertas_Minyak_Makanan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kertas_Minyak_Makanan</a>
	Kelas	Kemasan
Plastik	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Plastik">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Plastik</a>
	Kelas	Kemasan

Jalan	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jalan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jalan</a>
	Kelas	Lokasi
Kecamatan	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kemasan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kemasan</a>
	Kelas	Lokasi
Capcay	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Capcay">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Capcay</a>
	Kelas	Makanan
Gudeg	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gudeg">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gudeg</a>
	Kelas	Makanan
Kopi	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kopi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kopi</a>
	Kelas	Minuman
Susu	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu</a>
	Kelas	Minuman
Asin	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Asin">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Asin</a>
	Kelas	Rasa
Pahit	URI	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pahit">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pahit</a>
	Kelas	Rasa

### 3.2.5 Deskripsi Anotasi

Penamaan menggunakan anotasi pada ontologi diperlukan dikarenakan penamaan langsung pada *instance*, *Object Properties*,

dan *data properties* tidak mendukung adanya spasi. Maka dari itu untuk penamaan langsung kepada 3 properti diatas, bila penamaan nya lebih dari 2 kata, maka antara kata pertama dan kata kedua tidak boleh dipisah, atau bisa juga di pisahkan dengan tanda ( \_ ) dan ( - ).

Maka, pada pemberian anotasi kali ini, fokus utamanya adalah bagaimana nama dari ketiga properti diatas dapat terlihat lebih rapih dan lebih enak dipandang. Walaupun semua properti akan diberikan anotasi, namun untuk properti yang namanya hanya terdiri dari 1 kata saja, penamaan anotasi nya akan sama/mengikuti nama properti nya. Untuk properti yang memiliki nama lebih dari 2 kata, maka pada nama anotasi nya akan dihilangkan tanda ( - ) atau tanda ( \_ ) dan akan diganti dengan spasi antar kata nya. Penambahan anotasi menggunakan fungsi *rdfs:label* yang sudah disediakan.

Penambahan anotasi khususnya pada *instance* dapat bersifat *one to many*, yang berarti untuk 1 *instance* dapat memiliki lebih dari 1 anotasi. Sebagai contoh untuk mendeskripsikan *instance* merica pada kelas Bahan. Merica dan Lada adalah 2 nama yang sama. Maka dari itu untuk nama *instance* Lada/Merica akan memiliki 3 anotasi, yaitu : Lada, Merica dan Lada/Merica.

Tabel 3.11 akan mendeskripsikan beberapa contoh *instance* yang mengalami penambahan spasi pada nama anotasi nya.

**Tabel 3.6 Beberapa contoh *instance* sebelum dan sesudah ditambahkan anotasi**

<b><i>Instance</i> sebelum ditambahkan Anotasi</b>	<b><i>Instance</i> setelah ditambahkan anotasi</b>
Cabai_Merah	Cabai Merah
Baking_Powder	Baking Powder
Daging_Sapi	Daging Sapi
Bumbu_Kacang_Sate_Ayam	Bumbu Kacang Sate Ayam
Tumis_Kacang_Panjang	Tumis Kacang Panjang
Telur_Ayam_Rebus	Telur Ayam Rebus
Sayur_Tempe_Tahu	Sayur Tempe Tahu

Kertas_Minyak_Makanan	Kertas Minyak Makanan
Lemper	Lemper
	Lemper Ayam
Lada/Merica	Lada
	Merica
	Lada/Merica

### 3.2.6 Deskripsi *Rule*

Pada tahap ini dilakukan perancangan *Rule* Menggunakan SWRL (Semantic Web Rule Language). *Rule* dirancang untuk mencari apakah ada pengetahuan yang bersifat tersirat. Hasil yang diharapkan dari perancangan *rule* adalah agar muncul nya fakta-fakta baru yang bermanfaat. Penulisan *query* pada *rule* mengacu pada kelas, relasi, atribut, dan individu.

Pada ontologi yang dibangun, terdapat 1 *rule* yang dirancang. *Rule* ini digunakan untuk mencari hubungan antara bahan yang terkandung dengan suatu makanan dengan rasa dari suatu makanan tersebut.

```
foo:Foody(?f) ^ foo:Bahan(?b) ^ foo:Rasa(?r) ^ foo:Mengandung(?f, ?b) ^ foo:Memiliki_Rasa(?b, ?r) -> foo:Memiliki_Rasa(?f, ?r)
```

Berdasarkan *rule* tersebut, dapat disimpulkan bahwa bila suatu makanan yang mengandung suatau bahan, dimana bahan tersebut memiliki rasa A, maka otomatis makanan yang mengandung bahan tersebut juga akan memiliki rasa A.

### 3.2.7 Skenario Uji Coba

Setelah kelas, property, dan *instance* dibangun, selanjutnya adalah dilakukan perancangan studi kasus untuk scenario menguji ontologi dengan menggunakan Bahasa SPARQL. Studi kasus ini dirancang salah satu nya untuk menguji dari ontologi yang sudah

dirancang. Hasil yang diharapkan adalah dapat untuk melihat property, ataupun *instance* yang diminta sesuai dengan studi kasus. Penulisan query pada studi kasus mengacu kepada kelas, properti objek, properti data, dan anotasi pada ontologi.

Terdapat 2 jenis skenario uji coba. Pertama, adalah uji coba dengan menggunakan sintaks yang benar, dan yang kedua adalah uji coba dengan menggunakan sintaks yang salah. Harapannya adalah skenario ini dapat menunjukkan bagaimana sistem akan bertindak terhadap input yang masuk, baik itu benar maupun salah.

Semua query pada studi kasus ini dijalankan dengan menggunakan prefix yang sama sebagai berikut :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foo: <http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#>
```

**Kode Sumber 3.1 Prefix yang digunakan pada query**

### 3.2.7.1 Uji Coba Dengan Sintaks Yang Benar

Pada uji coba dengan sintaks yang benar, terdapat 10 studi kasus yang bisa dijadikan pengujian. Subbab berikutnya akan menjelaskan tiap-tiap studi kasus beserta sintaksnya.

#### 3.2.7.1.1 Studi Kasus 1 : Menampilkan Bahan yang dipakai untuk Makanan dengan cara penyajian di Rebus

```
SELECT DISTINCT ?name
WHERE {
```

```

?Bahan foo:DiRebus ?Makanan.
?Bahan rdfs:label ?name .
}

```

### Kode Sumber 3.2 Sintaks SPARQL studi kasus 1

Pada studi kasus 1, yang akan menjadi *Output* adalah list dari *instances* pada kelas bahan yang memiliki relasi DiRebus kepada *instance* pada kelas Makanan. Variabel `rdfs:label` diatas diperuntukkan untuk mengakses nama anotasi dari *instance* pada kelas makanan.

#### 3.2.7.1.2 Studi Kasus 2 : Menampilkan Makanan yang dikemas menggunakan Daun Pisang

```

SELECT DISTINCT ?nama
WHERE {
  ?Makanan foo:DiKemas ?Kemasan.
  ?Makanan foo:DiKemas foo:Daun_Pisang.
  ?Makanan rdfs:label ?nama.
}

```

### Kode Sumber 3.3 Sintaks SPARQL studi kasus 2

Sintaks pada studi kasus 2 mengidentifikasi *instance* dari kelas makanan yang memiliki relasi DiKemas kepada *instance* Daun Pisang pada kelas Kemasan. *Output* dari studi kasus ini adalah mencoba mencari makanan apa saja yang dikemas dengan menggunakan daun pisang.

#### 3.2.7.1.3 Studi Kasus 3 : Menampilkan kandungan bahan pada makanan capcay

```

SELECT DISTINCT ?nama
WHERE {
  ?Bahan foo:DiTumis ?Makanan.
  ?Bahan foo:DiTumis foo:Capcay.
}

```



```

        ?Bahan rdfs:label ?nama.
    }

```

### Kode Sumber 3.4 Sintaks SPARQL studi kasus 3

Pada Studi kasus 10, *Output* nya adalah list bahan apa saja yang terkandung pada makanan capcay. Pertama akan dicari terlebih dahulu *instance* capcay pada makanan. Lalu selanjutnya akan dicari *instance* dari kelas bahan apa saja yang terhubung kepada *instance* capcay melalui relasi DiTumis.

#### 3.2.7.1.4 Studi Kasus 4 : Menampilkan Minuman yang Memiliki Rasa Pahit

```

SELECT DISTINCT ?nama
WHERE {
    ?Minuman foo:Memiliki_Rasa ?Rasa.
    ?Minuman foo:Memiliki_Rasa foo:Pahit.
    ?Minuman rdfs:label ?nama.
}

```

### Kode Sumber 3.5 Sintaks SPARQL studi kasus 4

Pada studi kasus 4, langkah yang akan ditempuh adalah menampilkan *instance* pada kelas minuman yang memiliki relasi memiliki rasa kepada *instance pahit* pada kelas rasa. *Output* dari studi kasus ini adalah nama minuman yang memiliki rasa pahit.

#### 3.2.7.1.5 Studi Kasus 5: Menampilkan Bahan yang bila dipanggang akan menghasilkan makanan yang memiliki rasa gurih

```

SELECT DISTINCT ?nama1
WHERE {
    ?Bahan foo:DiPanggang ?Makanan.
    ?Makanan foo:Memiliki_Rasa ?Rasa.
}

```

```

    ?Makanan foo:Memiliki_Rasa foo:Gurih.
    ?Bahan rdfs:label ?nama1.
}

```

**Kode Sumber 3.6 Sintaks SPARQL studi kasus 5**

Ouput dari studi kasus ke 5 ini adalah, menampilkan list *instance* dari kelas Bahan yang memiliki relasi DiPanggang kepada kelas Makanan, lalu *instance* pada kelas Makanan tersebut, memiliki relasi Memiliki Rasa kepada *instance* Gurih yang berada di kelas Rasa.

### 3.2.7.1.6 Studi Kasus 6 : Menampilkan Makanan yang Memiliki Rasa Pedas

```

SELECT DISTINCT ?nama
WHERE {
    ?Makanan foo:Memiliki_Rasa ?Rasa.
    ?Makanan foo:Memiliki_Rasa foo:Pedas.
    ?Makanan rdfs:label ?nama.
}

```

**Kode Sumber 3.7 Sintaks SPARQL studi kasus 6**

Pada studi kasus 6, tujuan nya adalah untuk menampilkan individu dari kelas makanan, yang memiliki relasi memiliki rasa ke kelas rasa dan menunjuk kepada individu pedas. *Output* dari studi kasus ini adalah *list* dari makanan yang memiliki rasa pedas.

### 3.2.7.1.7 Studi Kasus 7 : Menampilkan Bahan yang digunakan sebagai pelengkap dari suatu makanan beserta nama makanan terkait

```

SELECT DISTINCT ?nama1 ?nama2
  WHERE { ?Bahan foo:Sebagai_Pelengkap ?Makanan;
          rdfs:label ?nama1.
          ?Makanan rdfs:label ?nama2.
        }

```

**Kode Sumber 3.8 Sintaks SPARQL studi kasus 7**

Pada studi kasus 7, langkah yang akan ditempuh adalah menampilkan instance pada kelas bahan yang memiliki relasi Sebagai Pelengkap kepada instance pada kelas makanan. *Output* dari studi kasus ini adalah nama Bahan yang digunakan sebagai pelengkap dari suatu makanan disertai dengan Makanan nya.

Di akhir dari baris ke 2 dari sintaks diatas, terdapat tanda “;”. Tanda tersebut digunakan bila baris sintaks selanjutnya memiliki subjek yang sama dengan baris diatasnya. Maka subjek pada baris ke 3 tidak perlu ditulis kembali karena pada baris ke 2 sudah diberi tanda “;” pada akhir baris.

### 3.2.7.1.8 Studi Kasus 8 : Menampilkan Makanan yang Dikemas Dengan Plastik

```

SELECT DISTINCT ?nama
  WHERE {
          ?Makanan foo:DiKemas ?Kemasan.
          ?Makanan foo:DiKemas foo:Plastik.
          ?Makanan rdfs:label ?nama.
          ?Makanan rdf:type foo:Makanan.
        }

```

**Kode Sumber 3.9 Sintaks SPARQL studi kasus 8**

Pada studi kasus 8, tujuannya adalah untuk menampilkan individu dari kelas makanan, yang memiliki relasi dikemas ke kelas kemasan dan menunjuk kepada individu plastik. *Output* dari studi kasus ini adalah *list* dari makanan yang dikemas dengan plastik.

### 3.2.7.1.9 Studi Kasus 9 : Menampilkan Kemasan yang digunakan untuk sate ayam

```
SELECT DISTINCT ?nama
WHERE {
    ?Makanan foo:DiKemas ?Kemasan.
    foo:Sate_Ayam foo:DiKemas ?Kemasan.
    ?Kemasan rdfs:label ?nama.
}
```

#### Kode Sumber 3.10 Sintaks SPARQL studi kasus 9

Pada studi kasus 9, terdapat *instance* dari kelas makanan yaitu Sate Ayam, lalu memiliki relasi DiKemas kepada banyak *instance* yang berada pada kelas Kemasan. Ouput dari studi kasus ini adalah menampilkan *instance* dari kelas Kemasan yang memiliki relasi DiKemas dengan Sate Ayam.

### 3.2.7.1.10 Studi Kasus 10 : Menampilkan Minuman yang Memiliki Rasa Manis

```
SELECT DISTINCT ?nama
WHERE {
    ?Minuman foo:Memiliki_Rasa ?Rasa.
    ?Minuman foo:Memiliki_Rasa foo:Manis.
    ?Minuman rdfs:label ?nama.
    ?Minuman rdf:type foo:Minuman.
}
```

#### Kode Sumber 3.11 Sintaks SPARQL studi kasus 10

Pada studi kasus ke 10, terdapat banyak *instance* dari kelas makanan dan minuman, lalu memiliki relasi memiliki rasa kepada *instance* Manis pada kelas Rasa. Ouput dari studi kasus ini adalah

menampilkan *instance* dari kelas makanan dan minuman yang memiliki rasa yang Manis.

### 3.2.7.2 Uji Coba Denga Sintaks yang Salah

Pada uji coba dengan sintaks yang salah, terdapat 5 studi kasus yang bisa dijadikan pengujian. Pada studi kasus yang salah, penomoran pada setiap studi kasus menggunakan huruf “S” yang memiliki singkatan dari kata “salah” dan memiliki arti bahwa studi kasus ini menggunakan sintaks yang memang disengaja salah. Subbab berikut nya akan menjelaskan tiap-tiap studi kasus beserta sintaks nya.

#### 3.2.7.2.1 Studi Kasus S1 : Menampilkan Bahan yang digunakan sebagai pelengkap dari suatu makanan beserta nama makanan nya

```
SELECT DISTINCT ?nama1 ?nama2
WHERE { ?Bahan foo:Sebagai_Pelengkap ?Makanan;
        rdfs:label ?nama1.
        ?Makanan rdfs:label ?nama1.
}
```

**Kode Sumber 3.12 Sintaks SPARQL studi kasus S1**

Tujuan dari studi kasus S1 ini adalah, menampilkan *instance* dari kelas bahan yang memiliki relasi sebagai pelengkap dari suatu *instance* dari kelas makanan beserta nama *instance* nya. Kesalahan pada studi kasus ini ada pada baris ke 4 dari kode.

#### 3.2.7.2.2 Studi Kasus S2 : Menampilkan Kandungan Bahan pada Makanan Capcay

```

SELECT DISTINCT ?nama
WHERE {
    ?Bahan foo:DiTumis ?Makanan.
    ?Bahan foo:DiTumis foo:Capcay.
    ?Bahan rdfs:label ?nama.
}

```

**Kode Sumber 3.13 Sintaks SPARQL studi kasus S2**

Pada Studi kasus S2, *Output* nya adalah list bahan apa saja yang terkandung pada makanan capcay. Pertama akan dicari terlebih dahulu *instance* capcay pada makanan. Lalu selanjutnya akan dicari *instance* dari kelas bahan apa saja yang terhubung kepada *instance* capcay melalui relasi DiTumis. Sebenarnya, studi kasus ini sama dengan studi kasus benar nomor 3, namun disini sengaja diberikan kesalahan kode pada baris ke 5.

### 3.2.7.2.3 Studi Kasus S3 : Menampilkan Makanan untuk Porsi Individu

```

SELECT DISTINCT ?nama
WHERE {
    ?Makanan foo:Porsi ?jml.
    ?Makanan rdfs:label ?nama.
    FILTER (?jml='Individu').
}

```

**Kode Sumber 3.14 Sintaks SPARQL studi kasus S3**

Pada studi kasus S3, ingin menampilkan makanan apa saja yang memiliki porsi untuk individu. Namun, kesalahan pada baris ke 3 membuat studi kasus ini tidak bisa mengeluarkan hasil sama sekali.

### 3.2.7.2.4 Studi Kasus S4 : Menampilkan Bahan yang Dipakai untuk Makanan dengan Penyajian di Rebus

```
SELECT DISTINCT ?Bahan
WHERE {
  ?Bahan foo:DiRebus ?Makanan.
  ?Bahan rdfs:label ?name .
}
```

Kode Sumber 3.15 Sintaks SPARQL studi kasus S4

Pada studi kasus S4, *Output* nya adalah menampilkan list dari *instance* bahan yang memiliki relasi DiRebus dengan *instance* pada kelas makanan. Namun, kesalahan pada baris 1 membuat hasil dari studi kasus ini bukan menampilkan nama dari *instance* terkait, melainkan menampilkan Uniform Resource Identifier (URI) dari tiap *instance*.

### 3.2.7.2.5 Studi Kasus S5 : Menampilkan Makanan yang Memiliki Rasa yang Pedas.

```
SELECT DISTINCT ?nama
WHERE { ?Makanan foo:Memiliki_Rasa ?Rasa.
        ?Makanan foo:Memiliki_Rasa foo:Pedas.
}
```

Kode Sumber 3.16 Sintaks SPARQL studi kasus S5

Pada studi kasus S5, kesalahan pada baris 1 membuat studi kasus ini tidak bisa menampilkan *Output* yang diharapkan.

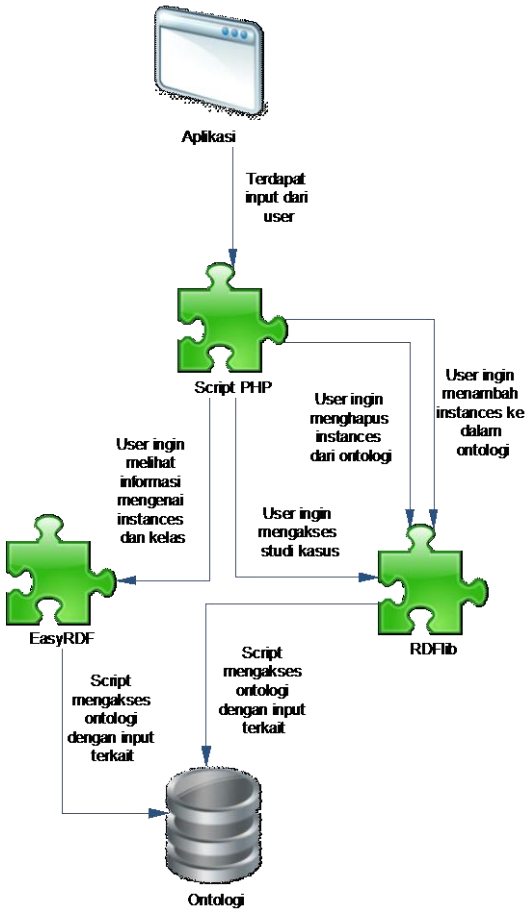
## 3.3 Analisis dan Perancangan Sistem

### 3.3.1 Deskripsi Umum Sistem

Pada subbab ini akan dijelaskan lebih detail mengenai aplikasi yang akan dibangun pada tugas akhir ini. Nama aplikasinya adalah Foody. Nama Foody diambil dari kalimat Food Identity dan disingkat menjadi Foody.

### 3.3.1.1 Arsitektur Sistem

Gambar 3.3 akan menjelaskan arsitektur dari sistem ini





### Gambar 3.3 Diagram Arsitektur dari sistem

Pada sistem Foody, terdapat beberapa fitur yang bisa dilakukan oleh *user*. beberapa fitur tersebut adalah sebagai berikut :

1. *User* dapat memilih lalu melihat informasi dari *instance* yang berada pada ontologi. List dari *instance* yang bisa dilihat akan ditampilkan pada menu dropdown yang disediakan pada halaman utama dari sistem.
2. *User* dapat melihat studi kasus pada ontologi. Disediakan halaman khusus bagi *user* untuk melihat studi kasus beserta *Output* dari studi kasus nya
3. *User* dapat menambah *instance* baru ke dalam ontologi. Terdapat halaman khusus untuk *user* menginputkan nama kelas nya beserta nama *instance* nya
4. *User* dapat menghapus *instance* yang ada di dalam ontologi. Ketika sedang melihat informasi dari *instance*, terdapat tombol hapus yang bisa di akses *user* untuk menghapus *instance* terkait.

Pada sistem ini, terdapat beberapa *Library* yang digunakan. *Library* pertama adalah EasyRDF. EasyRDF adalah *Library* PHP yang digunakan untuk mengakses ontologi. Pada sistem ini, EasyRDF digunakan untuk *user* agar dapat melihat informasi mengenai *instance* yang ada pada ontologi.

*User*, ketika ingin mengakses ontologi, dapat melalui halaman utama dari sistem. Pada halaman utama, telah tersedia menu yang berisi list-list *instance* pada dropdown select yang tersedia. Ketika *user* sudah memilih *instance* mana yang ingin dilihat informasi nya, maka hal ini akan menjadi input bagi EasyRDF. Selanjutnya, EasyRDF akan mengakses ontologi untuk mencari *instance* yang sesuai dengan input dari *user*. Setelah EasyRDF menemukan *instance* yang sesuai, maka EasyRDF akan menampilkan *instance*

tersebut ke dalam sebuah halaman HTML yang dapat dilihat oleh *user*.

Informasi *instance* yang akan ditampilkan di dalam halaman HTML berupa nama dari *instance* itu sendiri, nama anotasi, object properties (relasi) yang dipunya dari *instance* tersebut, serta nama *instance* lain yang memiliki relasi dengan *instance* tersebut. *User* juga dapat melihat informasi dari *instance* yang memiliki relasi dengan *instance* tersebut dengan cara mengklik nama *instance* yang ingin dilihat ketika informasi sedang disajikan. Maka EasyRDF selanjutnya akan menampilkan ulang informasi *instance* yang ingin dilihat oleh *user*.

*Library* kedua yang digunakan pada sistem ini adalah RDFlib. RDFlib adalah sebuah *Library* python untuk memudahkan dalam mengakses dan memanipulasi ontologi. Penggunaan *Library* RDFlib digunakan untuk beberapa fitur pada sistem ini diantaranya untuk melihat studi kasus, dan memanipulasi ontologi (menambah dan menghapus *instance* ke dalam ontologi).

Pada fitur melihat studi kasus, *user* dapat memilih ingin melihat studi kasus nomor berapa yang sudah disediakan di dalam menu dropdown. Ketika *user* memilih studi kasus nomor berapa, maka selanjutnya hal ini akan memicu script PHP dari tiap-tiap studi kasus. Selanjutnya, script php tadi akan memanggil script python yang berisi *Library* RDFlib untuk melakukan query ke dalam ontologi. Hasil dari query yang dilakukan oleh RDFlib akan ditampilkan ke dalam halaman HTML.

Pada fitur menambah *instance* baru ke dalam ontologi, *user* dapat menginputkan nama *instance* yang ingin dimasukkan beserta nama kelas nya di dalam sebuah form yang sudah disediakan. Ketika *user* mengklik tombol submit. Maka selanjutnya data yang *user* isi pada form tadi akan di parsing ke dalam script PHP. Selanjutnya, script PHP akan memanggil script Python sekaligus memparsing data input ke dalam script python yang didalamnya ada *Library* RDFlib. Script Python ini yang nantinya akan mengakses ontologi, lalu memasukkan *instance* baru ke dalam kelas berdasarkan data input dari *user*. ketika proses sudah selesai.

*User* akan diarahkan ke halaman utama. Dan selanjutnya *user* dapat mengecek apakah *instance* yang diisikan sudah berhasil masuk atau sebaliknya.

Pada fitur menghapus *instance*, *user* dapat melakukan ini ketika sedang melihat informasi dari *instance*. Terdapat tombol hapus yang dapat di klik oleh *user*. ketika hal ini dilakukan oleh *user*, maka sistem akan mengambil data mengenai *instance* apa yang akan di hapus, lalu memarsing data nya ke script PHP. Selanjutnya script PHP akan memanggil script Python sekaligus memarsing data nya. Script Python, yang didalam nya terdapat *Library* RDFlib, ini yang nantinya akan mengakses ontologi lalu menghapus *instance* terkait.

*[Halaman ini sengaja dikosongkan]*

## **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan sistem yang telah dibahas sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fitur-fitur yang diimplementasikan dalam program Tugas Akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

### **4.1 Lingkungan Implementasi Perangkat Lunak**

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

<b>Perangkat</b>	<b>Jenis Perangkat</b>	<b>Spesifikasi</b>
<b>Perangkat Keras</b>	Prosesor	Intel(R) Core(TM) i7-6700HQ CPU @2.60GHz
	Memori	16 GB
<b>Perangkat Lunak</b>	Sistem Operasi	Windows 10 Pro 64-bit (Build 17134)
	Perangkat Pengembang	Apache Server 2.0

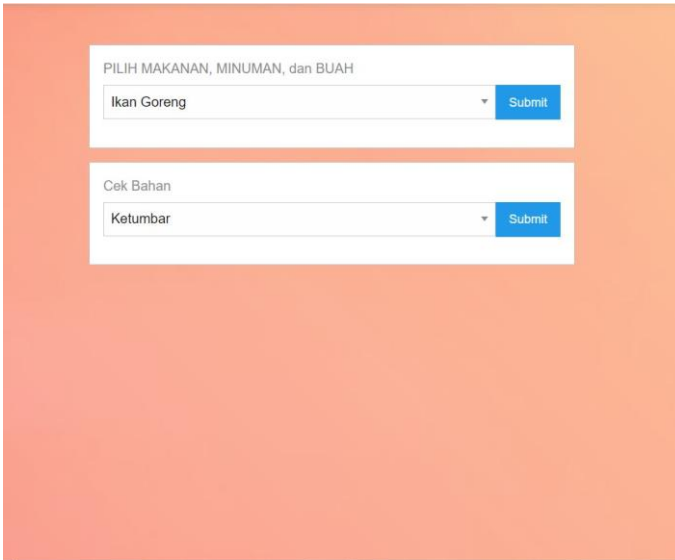
### **4.2 Implementasi**

Pada subbab ini akan dijelaskan bagaimana cara membangun perangkat lunak ini secara detail dan menampilkan kode sumber

serta langkah-langkah yang dilakukan pada semua tahapan pembuatan perangkat lunak

#### 4.2.1 Implementasi Fitur Melihat *Instances* Pada Ontologi

Fitur ini memungkinkan *user* untuk dapat memilih *instance* dan melihat informasi dari *instance* yang dipilih. Gambar 4.1 menunjukkan halaman dari implementasi fitur ini



The image shows a web interface with two sections. The first section is titled "PILIH MAKANAN, MINUMAN, dan BUAH" and contains a dropdown menu with "Ikan Goreng" selected and a blue "Submit" button. The second section is titled "Cek Bahan" and contains a dropdown menu with "Ketumbar" selected and a blue "Submit" button.

**Gambar 4.1** Halaman Implementasi Fitur Melihat *Instance*

Isi dari dropdown yang terdapat pada halaman diatas adalah list dari nama-nama *instance* dari ontologi yang dapat dilihat oleh *user*. *Library* yang digunakan pada fitur ini adalah EasyRDF. Dropdown pertama yang berada di atas, adalah dropdown untuk menampilkan list *instance* yang berada pada kelas Foody. Sedangkan dropdown kedua yang berada di bawah, adalah

dropdown untuk menampilkan *instance* yang berada pada kelas Bahan. Kode sumber 4.1 dan 4.2 akan menunjukkan kode program dari kedua dropdown tersebut

```

1. <?php
2.         foreach($graph-
   >allOfType('foo:Foody') as $name) {
3.             echo "<option value='". $name
   ."'>".str_replace('http://www.semanticweb.org
   /asus/ontologies/2019/1/untitled-ontology-
   49#', "", $name-
   >get('rdfs:label'))."</option>";
4.             echo $name;
5.         }
6.
7. ?>

```

**Kode Sumber 4.1 Dropdown Untuk Kelas Foody**

```

8. <?php
9.         foreach($graph-
   >allOfType('foo:Bahan') as $name) {
10.            echo "<option value='". $name.
   "'>".str_replace('http://www.semanticweb.org/a
   sus/ontologies/2019/1/untitled-ontology-
   49#', "", $name-
   >get('rdfs:label'))."</option>";
11.            echo $name;
12.        }
13.
14. ?>

```

**Kode Sumber 4.2 Dropdown Untuk Kelas Bahan**

Setelah *user* memilih *instances* mana, halaman sistem akan menampilkan informasi dari *instance* yang dipilih.

## 4.2.2 Implementasi Fitur Melihat Studi Kasus

Fitur ini memungkinkan *user* untuk dapat melihat beberapa skenario studi kasus yang dirancang untuk menguji ontologi. Pada fitur ini, menggunakan *Library RDFlib*. *User* dapat memilih studi kasus mana yang ingin dilihat dengan menggunakan menu dropdown. Gambar 4.3 akan menunjukkan lebih detail



**Gambar 4.2 Halaman Fitur Melihat Studi Kasus**

Setelah *user* memilih studi kasus mana yang akan dilihat, maka hal ini akan memanggil script PHP. Kode sumber 4.3 akan menunjukkan kode program dari script Python yang dipanggil.



```
1. <?php
2.
3. //$input = $_GET['inp'];
4.
5. $command = escapeshellcmd("C:\\Users\\ASUS\\AppData\\Local\\Programs\\Python\\Python36\\python.exe C:\\xampp\\htdocs\\foody\\query\\query1.py");
6. exec($command, $output);
7. //var_dump($output);
8. foreach($output as $each_output){
9. echo $each_output.'<br>';
10. }
11.
12. ?>
```

#### Kode Sumber 4.3 Script PHP Pada Implementasi Fitur Melihat Studi Kasus

Kegunaan dari script PHP diatas, adalah script diatas yang nanti nya akan memanggil script Python yang berisi *Library* RDFlib. Script Python inilah yang nanti nya akan mengakses ontologi, lalu pada fitur ini melakukan query, sehingga hasilnya nanti dapat dilihat oleh *user*. 1 script python dapat melakukan 1 query untuk 1 studi kasus. Pada sistem ini terdapat 10 studi kasus. Jadi terdapat 10 script python yang dapat dieksekusi. kode sumber 4.4 akan menunjukkan contoh script Python nya. Selanjutnya ketika proses query sudah selesai, maka hasil akhir akan ditampilkan di halaman HTML.

```

1. import rdflib
2.
3. g = rdflib.Graph()
4.
5. g.parse("foody.rdf")
6.
7. qres = g.query(
8.     """
9. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#>
10. PREFIX owl: <http://www.w3.org/2002/07/owl#>
11. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-
    schema#>
12. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
13. PREFIX foo: <http://www.semanticweb.org/asus/ontolog
    ies/2019/1/untitled-ontology-49#>
14. SELECT DISTINCT ?name
15. WHERE {
16.   ?Bahan foo:DiRebus ?Makanan.
17.   ?Bahan rdfs:label ?name .
18.   }"""
19.
20. for row in qres:
21.     print("%s" % row)

```

**Kode Sumber 4.4** Kode sumber dari Script Python untuk Implementasi Fitur Melihat Studi Kasus untuk query nomor 1

### 4.2.3 Implementasi Fitur Menambah *Instance* Ke Dalam Ontologi

Pada Fitur ini, *user* dapat menambah *instance* baru ke dalam ontologi. Pada fitur ini menggunakan *Library* RDFlib. *User* dapat mengisi kan data *instances* yang akan dimasukkan pada form yang sudah disediakan. Form nya akan ditunjukkan oleh gambar 4.5

Memasukkan instance baru

Isi data berikut.

Kelas

individu

Submit

**Gambar 4.3** Form Untuk Mengisi *Instance* Baru ke Dalam Ontologi

Setelah *user* sudah mengisi form dan men submit data nya. Maka data tersebut akan di parsing ke script PHP. Kode sumber 4.5 akan menunjukkan kode program dari script PHP tersebut

```

1. <?php
2. $kelas = $_POST['kelas'];
3. $individu = $_POST['individu'];
4.
5. $count = 3;
6. $pass = "[".$kelas."',".$individu."]";
7.
8. $command = escapeshellcmd("C:\\Users\\ASUS\\AppData\\Local\\Programs\\Python\\Python36\\python.exe C:\\xampp\\htdocs\\foody\\insert.py \"$pass\" \"$count\" ");
9. exec($command, $output);
10.
11. header('Location: index.php');
12.
13. ?>

```

**Kode Sumber 4.5** Script PHP untuk Fitur Menambah *Instance*

Kegunaan dari script PHP tersebut adalah untuk memanggil Script Python. Nantinya, script Python, yang didalamnya terdapat *Library* RDFlib ini yang akan mengakses ontologi, lalu akan menambah *instance* baru sesuai dengan input dari *user*. Data mengenai *instance* baru ini juga nanti nya akan di parsing ke script Python. Kode sumber 4.6 akan menunjukkan script Python untuk fitur menambah *instance*.

```

1. g = Graph()
2. g.parse("C:\\xampp\\htdocs\\foody\\foody.rdf")
3.
4. var_kelas = URIRef("http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#+kelas")
5. var_individu = URIRef("http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#+individu")
6.
7. g.add( (var_individu, RDF.type, var_kelas ) )
8. g.add( (var_individu, RDFS.label, Literal(individu , lang="en")) )
9.
10. g.serialize("C:\\xampp\\htdocs\\foody\\foody.rdf")

```

**Kode Sumber 4.6 Script Python untuk Fitur Menambah *Instance***

#### 4.2.4 Implementasi Fitur Menghapus *Instance* dari Ontologi

```

1. <?php
2. $entity = $_GET['entity'];
3.
4. $count = 3;
5. $pass = "[".$entity."]";
6.
7. $command = escapeshellcmd("C:\\Users\\ASUS\\AppData\\Local\\Programs\\Python\\Python36\\python.exe C:\\xampp\\htdocs\\foody\\delete.py \"$pass\" \"$count\" ");
8. exec($command, $Output);

```

```

9.
10. header('Location: index.php');
11.
12. ?>

```

#### **Kode Sumber 4.7 Script PHP untuk fitur Menghapus *Instance***

Pada Fitur ini, *User* dapat menghapus *instance* dari ontologi. Ketika informasi dari sebuah *instance* ditampilkan dengan menggunakan *Library* EasyRDF, *user* cukup mengklik tombol “Hapus” yang terdapat di bawah nama dari *instance* yang ditampilkan. Untuk fitur menghapus ini menggunakan *Library* RDFlib. Setelah *user* mengklik tombol “Hapus” maka selanjutnya data mengenai *instance* yang akan dihapus akan di parsing ke sebuah Script PHP. Kode sumber 4.7 akan menunjukkan kode program dari Script PHP tersebut.

Kegunaan dari script PHP ini adalah sama dengan script PHP pada fitur menambah *instance*, yaitu memanggil script Python yang didalamnya terdapat *Library* RDFlib, dan juga memarsing data mengenai *instance* yang akan dihapus. Script Python inilah yang nantinya akan mengakses ontologi untuk kemudian menghapus *instance* sesuai dengan data yang di parsing sebelumnya. Kode sumber 4.8 akan menunjukkan kode program dari script Python tersebut

```

1. g = Graph()
2. g.parse("C:\\xampp\\htdocs\\foody\\foody.rdf")
3.
4. entiti = URIRef("http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#+entity")
5.
6. g.remove( (entiti, None, None) )
7.
8. g.serialize("C:\\xampp\\htdocs\\foody\\foody.rdf")

```

#### **Kode Sumber 4.8 Script Python untuk Fitur Menghapus *Instance***

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk Tugas Akhir ini. Pengujian dilakukan pada Fitur dari Perangkat Lunak.

#### **5.1 Lingkungan Pengujian**

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam Tugas Akhir ini dilakukan pada setiap kasus penggunaan. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1.

**Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak**

<b>Perangkat</b>	<b>Spesifikasi</b>
Perangkat Keras	<b>Prosesor:</b> Intel® Core™ i7-6700 CPU @ 2.60GHz <b>Memori:</b> 16384MB RAM
Perangkat Lunak	<b>Sistem Operasi:</b> Windows 10 Home 64-bit <b>Perangkat Pengembang:</b> Python 3.6.8, PHP 7.2.11 <b>Library:</b> RDFlib 4.2.2, EasyRDF <b>Tools Pembantu:</b> Protégé

#### **5.2 Pengujian Fitur**

Pengujian pada sistem ini, menguji berdasarkan fitur yang sudah dibahas pada bab sebelumnya. Dikarenakan terdapat 4 fitur dari sistem, maka selanjutnya pengujian akan dibagi menjadi 4 subbab. Demi meningkatkan keabsahan dari pengujian, hasil dari

pengujian beberapa fitur, akan dibandingkan dengan bantuan dari tools pembantu protégé.

## 5.2.1 Fitur Melihat *Instance* dari Ontologi

Seperti yang sudah dibahas pada bab sebelumnya, fitur ini diperuntukkan untuk *user* agar dapat memilih *instances* dari ontologi, lalu melihat informasi dari *instance* yang dipilih.

Pengujian dilakukan dengan *user* memilih terlebih dahulu *instance* mana yang ingin dilihat. Ketika sudah memilih, *user* dapat mengklik tombol submit. Lalu informasi mengenai *instance* tersebut akan muncul di halaman tersebut. Gambar 5.1 akan menunjukkan hasil dari pengujian dan tabel 5.1 akan menunjukkan hasil lebih detail dari pengujian ini

The screenshot shows a web interface with two search forms at the top and a detailed view of an ontology instance below.

The first search form is titled "PILIH MAKANAN, MINUMAN, dan BUAH" and has a dropdown menu with "Ikan Goreng" selected and a "Submit" button.

The second search form is titled "Cek Bahan" and has a dropdown menu with "Ketumbar" selected and a "Submit" button.

The detailed view is titled "PENJELASAN" and "Ayam Bakar". It includes a "Hapus" link and a table with the following data:

Alias	Ayam Bakar
Rasa	Manis
Temperatur	Hangat
Kemasan	Plastik Kemasan Sterofoam Plastik
kandungan	Bawang Merah Margarin Kecap Manis Daging Ayam Garam

**Gambar 5.1** Hasil Pengujian Fitur Melihat *Instance* dari Ontologi



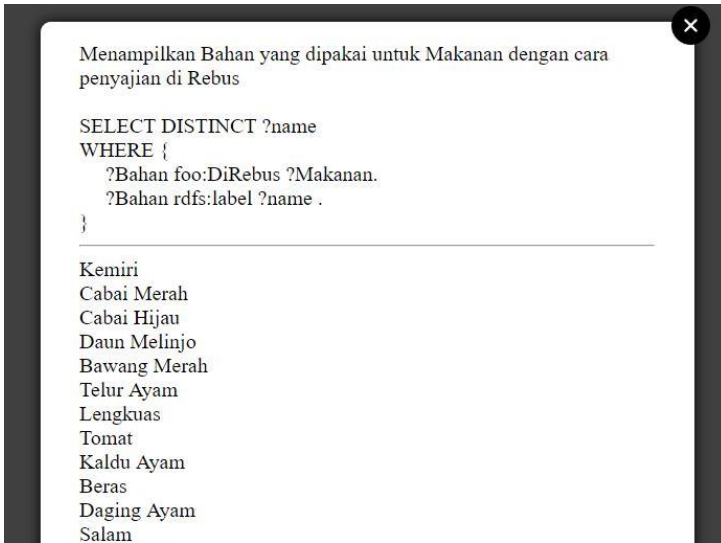
**Tabel 5.2 Pengujian Fitur Melihat *Instance* dari Ontologi**

<b>No. Pengujian</b>	P-001
<b>Nama</b>	Pengujian fitur melihat <i>instance</i> dari ontologi
<b>Tujuan Pengujian</b>	Menguji fungsionalitas untuk fitur melihat <i>instance</i> dari ontologi.
<b>Kondisi Awal</b>	<i>User</i> belum melihat <i>instance</i> yang diinginkan dari ontologi
<b>Langkah Pengujian</b>	<i>User</i> memilih <i>instance</i> .
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan <i>instance</i> yang dipilih beserta informasi dari <i>instance</i> tersebut.
<b>Hasil yang didapat</b>	Nama <i>instance</i> beserta informasi dari <i>instance</i> ditampilkan pada halaman sistem.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Nama <i>instance</i> dan informasi nya berhasil ditampilkan

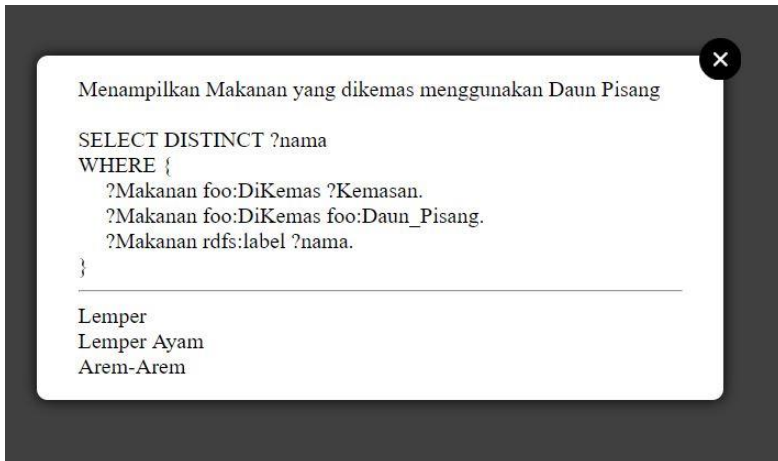
### 5.2.2 Fitur Melihat Studi Kasus

Fitur ini memungkinkan *user* untuk memilih studi kasus yang diinginkan, lalu melihat detail dari studi kasusnya berdasarkan hasil akhirnya.

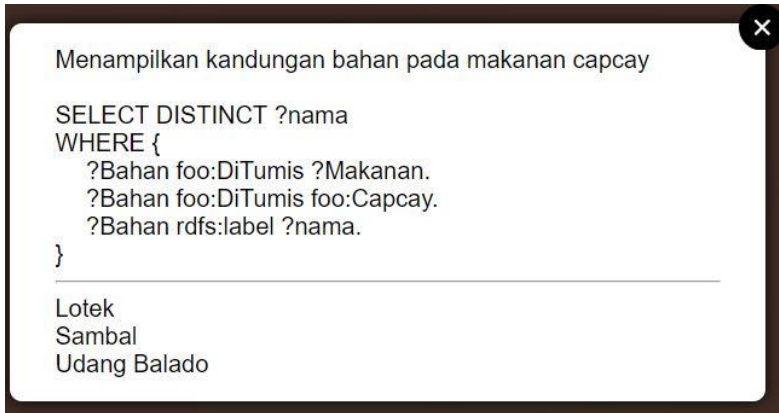
Pengujian dilakukan dengan *user* membuka halaman untuk melihat studi kasus. Lalu *user* dapat memilih studi kasus mana yang ingin dilihat. Dari 10 studi kasus, *user* hanya dapat melihat 1 studi kasus dalam 1 waktu. Ketika *user* sudah menentukan studi kasus yang mana yang akan ditampilkan, maka selanjutnya sistem akan menampilkan studi kasus yang dipilih oleh *user*. tabel 5.3 akan menjelaskan lebih detail mengenai pengujian pada fitur ini. Sedangkan gambar 5.2 sampai dengan gambar 5.12 akan menunjukkan hasil akhir dari pengujian fitur ini per studi kasus yang ada.



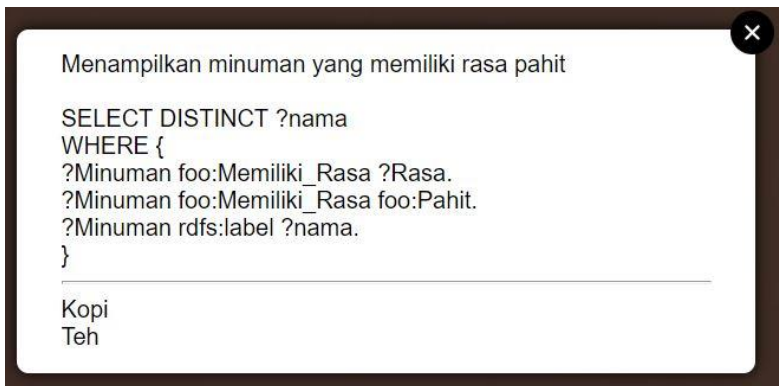
**Gambar 5.2 Hasil Pengujian Fitur Melihat Studi Kasus nomor 1**



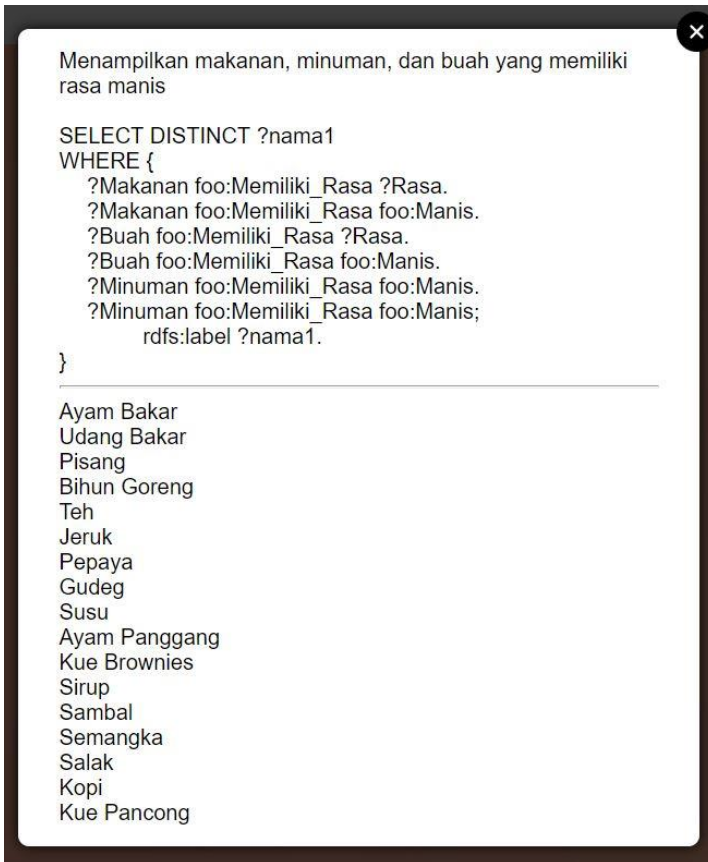
**Gambar 5.3 Hasil Pengujian Fitur Melihat Studi Kasus nomor 2**



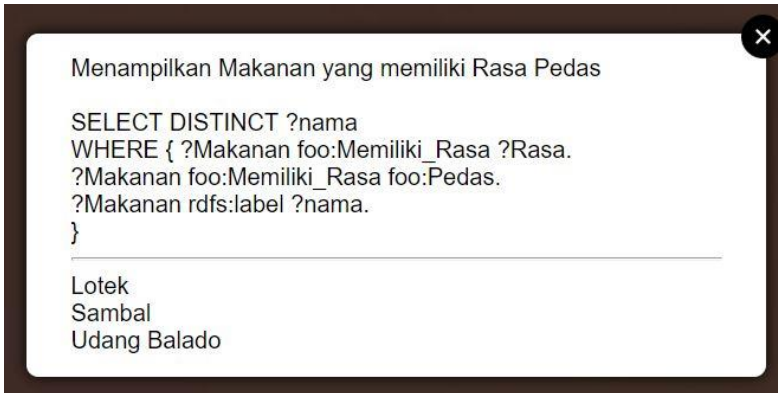
**Gambar 5.4 Hasil Pengujian Fitur Melihat Studi Kasus nomor 3**



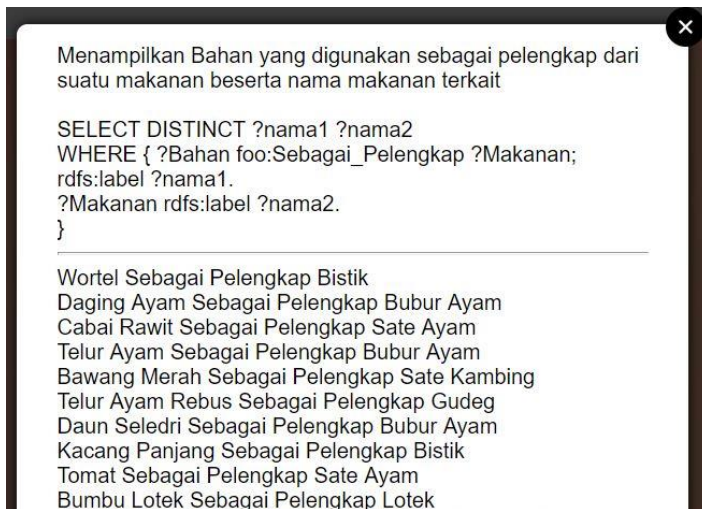
**Gambar 5.5 Hasil Pengujian Fitur Melihat Studi Kasus nomor 4**



**Gambar 5.6 Hasil Pengujian Fitur Melihat Studi Kasus nomor 5**



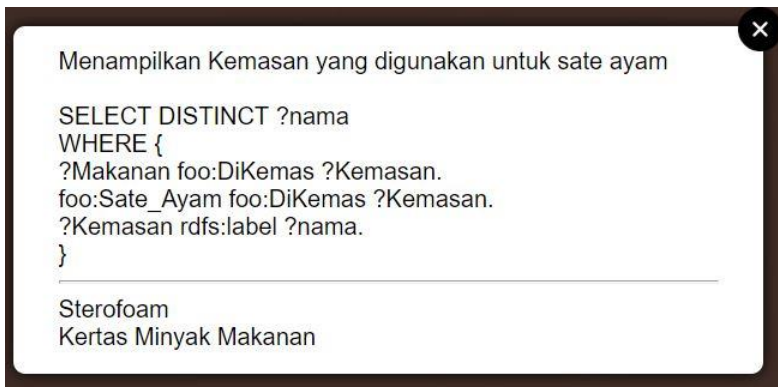
**Gambar 5.7 Hasil Pengujian Fitur Melihat Studi Kasus nomor 6**



**Gambar 5.8 Hasil Pengujian Fitur Melihat Studi Kasus nomor 7**



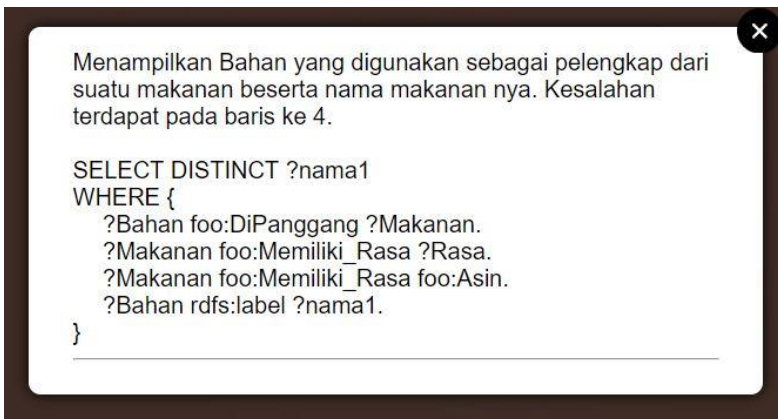
**Gambar 5.9 Hasil Pengujian Fitur Melihat Studi Kasus nomor 8**



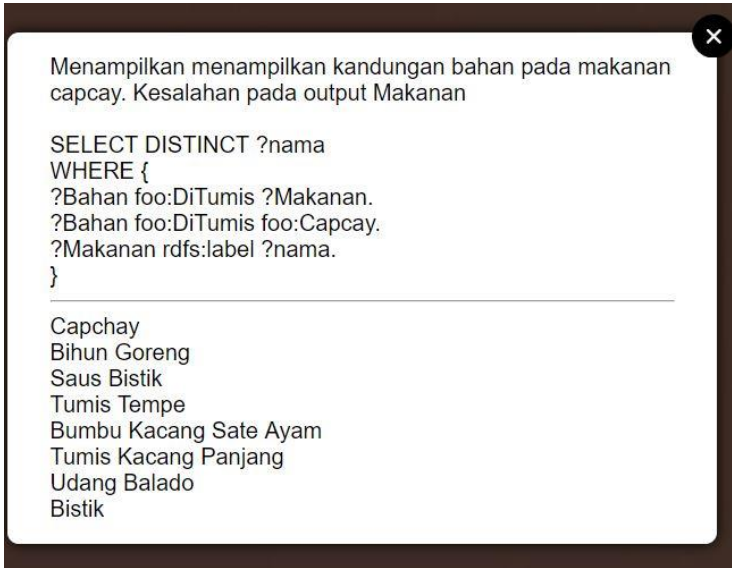
**Gambar 5.10 Hasil Pengujian Fitur Melihat Studi Kasus nomor 9**



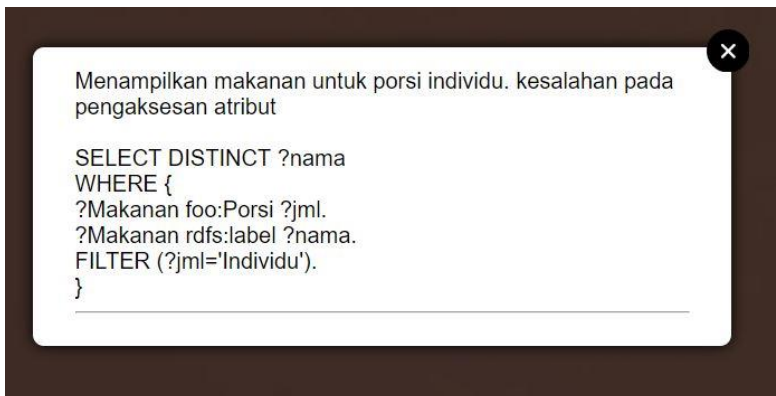
**Gambar 5.11 Hasil Pengujian Fitur Melihat Studi Kasus nomor 10**



**Gambar 5.12 Hasil Pengujian Fitur Melihat Studi Kasus nomor S1**

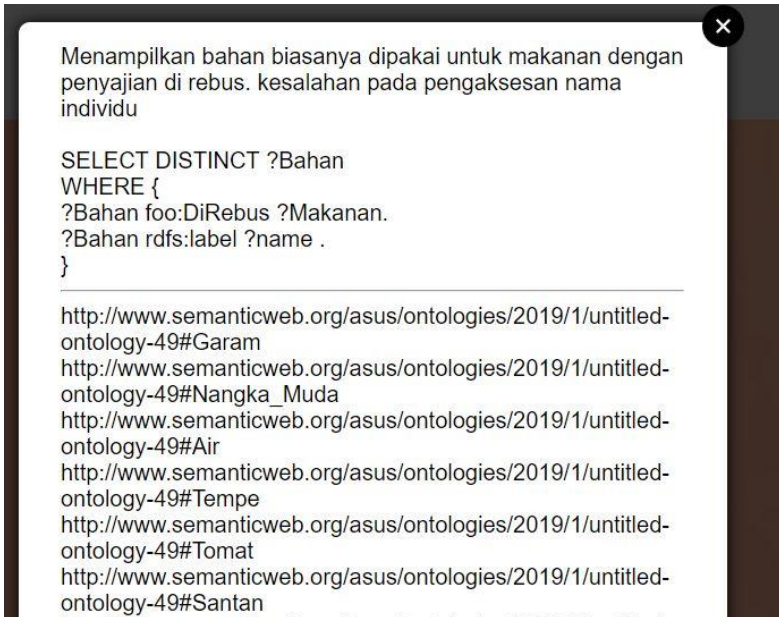


**Gambar 5.13 Hasil Pengujian Fitur Melihat Studi Kasus nomor S2**

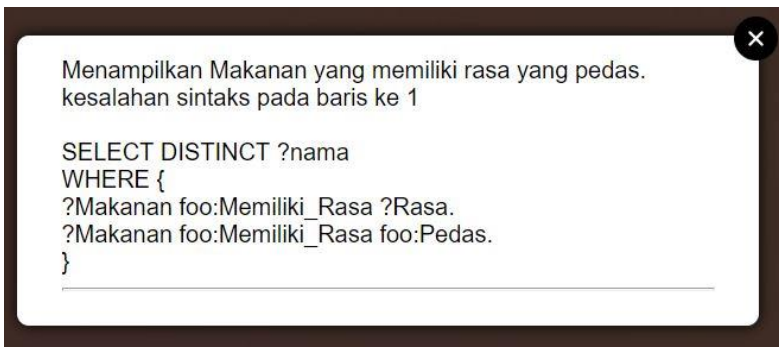


**Gambar 5.14 Hasil Pengujian Fitur Melihat Studi Kasus nomor S3**





**Gambar 5.15 Hasil Pengujian Fitur Melihat Studi Kasus nomor S4**



**Gambar 5.16 Hasil Pengujian Fitur Melihat Studi Kasus nomor S5**

**Tabel 5.3 Pengujian Fitur Melihat Studi Kasus**

<b>No. Pengujian</b>	P-002
<b>Nama</b>	Pengujian fitur melihat studi kasus.
<b>Tujuan Pengujian</b>	Menguji fungsionalitas untuk fitur melihat studi kasus.
<b>Kondisi Awal</b>	<i>User</i> belum melihat studi kasus.
<b>Langkah Pengujian</b>	<i>User</i> memilih studi kasus.
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan studi kasus yang dipilih oleh <i>user</i> .
<b>Hasil yang didapat</b>	Nama <i>instance</i> beserta informasi dari <i>instance</i> ditampilkan pada halaman sistem.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	Nama <i>instance</i> dan informasi nya berhasil ditampilkan

Pada pengukuran kinerja dari studi kasus diatas menggunakan confusion matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Rumus yang digunakan dalam pengukuran kinerja ini adalah :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

1. TP adalah True Positive, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
2. TN adalah True Negative, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
3. FN adalah False Negative, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.

4. FP adalah False Positive, yaitu jumlah data positif namun terklasifikasi salah oleh sistem.

Dari perhitungan dari pengujian ini didapatkan akurasi 95%. Berikut adalah perhitungan pengujian kinerja menggunakan confusion matrix:

$$Akurasi = \frac{15 + 5}{15 + 5 + 1 + 0} \times 100\%$$

***Akurasi= 95%***

Pada pengujian ini, terdapat 15 studi kasus yang dicoba yang terdiri dari 10 studi kasus benar dan 5 studi kasus salah. Studi kasus benar menggunakan sintaks yang memang dirancang untuk benar secara hasil dan sintaks, sedangkan studi kasus salah menggunakan sintaks yang disalahkan dan diharapkan hasilnya pun juga salah. Dari 15 studi kasus. 14 diantaranya memiliki hasil sesuai yang diharapkan. 9 studi kasus bernilai benar dan 5 sisanya bernilai salah. Namun ada 1 studi kasus yang diharapkan hasilnya benar, namun ternyata memiliki hasil yang tidak sesuai yang di harapkan. studi kasus tersebut adalah studi kasus benar nomor 10 sehingga studi kasus ini bernilai *True Negative*.

Pada studi kasus benar nomor 10, pada hasil akhir terdapat individu Kopi pada *list* minuman yang memiliki rasa manis. Padahal seharusnya, kopi memiliki rasa pahit. Namun pada ontologi ini, hal ini dapat terjadi karena pada minuman kopi, terdapat kandungan gula, walaupun kandungan gula pada minuman kopi tersebut tidaklah banyak sehingga tetap rasa pahit nya yang lebih dominan.

Maka dapat disimpulkan bahwa pemodelan data yang dibangun pada ontologi ini tidak memperhatikan banyak dan sedikit nya takaran pada suatu makanan. Selama makanan tersebut mengandung suatu bahan yang memiliki suatu rasa, maka makanan tersebut akan juga memiliki rasa.

### **5.2.3 Fitur Menambah *Instance* ke Dalam Ontologi**

Fitur ini memungkinkan *user* untuk menambah *instance* baru ke dalam ontologi. Pengujian dilakukan dengan memasukkan data dari form yang sudah disediakan. Pada pengujian kali ini, pengujian dilakukan dengan memasukkan *instance* dengan nama “Kambing Guling” ke dalam ontologi. Ketika *user* mengklik tombol submit. Maka akan diarahkan kembali ke halaman utama. Lalu *user* dapat mengecek apakah *instance* yang dimasukkan sudah ada pada list. Gambar 5.3, 5.4, dan tabel 5.4 akan menunjukkan detail dari pengujian fitur ini

The screenshot shows a web application interface with an orange background. At the top, there is a form titled "PILIH MAKANAN, MINUMAN, dan BUAH" with a dropdown menu containing "Ayam Panggang" and a blue "Submit" button. Below this is another form titled "Cek Bahan" with a dropdown menu containing "Coklat Batang" and a blue "Submit" button. At the bottom, there is a section titled "PENJELASAN" containing the text "Kambing Guling". Below the text is a "Hapus" link and a list of instances with the alias "Kambing Guling".

**Gambar 5.17 Hasil Pengujian Fitur Menambah *Instance* ke Ontologi**

**Tabel 5.4 Pengujian Fitur Menambah *Instance* ke Ontologi**

<b>No. Pengujian</b>	P-003
<b>Nama</b>	Pengujian fitur menambah <i>instance</i> ke ontologi.
<b>Tujuan Pengujian</b>	Menguji fungsionalitas untuk fitur menambah <i>instance</i> ke dalam ontologi.
<b>Kondisi Awal</b>	<i>User</i> belum menambah <i>instance</i> .
<b>Langkah Pengujian</b>	<i>User</i> memasukkan data mengenai <i>instance</i> yang ingin dimasukkan.

<b>Hasil yang diharapkan</b>	Sistem dapat memasukkan <i>instance</i> baru ke dalam ontologi berdasarkan input an dari <i>user</i> .
<b>Hasil yang didapat</b>	Nama <i>instance</i> baru berhasil dimasukkan ke dalam ontologi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	<i>Instance</i> baru sudah masuk ke dalam ontologi dan dapat dilihat oleh <i>user</i>



**Gambar 5.18 Hasil Pengujian Fitur Menambah *Instance* ke Ontologi**

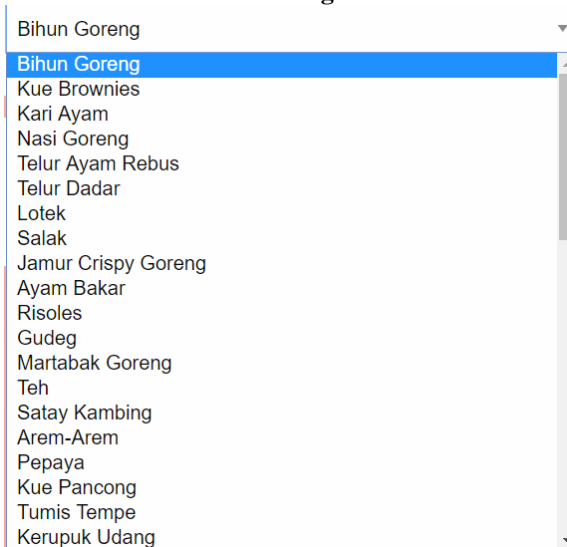
## 5.2.4 Fitur Menghapus *Instance* dari Ontologi

Fitur ini memungkinkan *user* untuk menghapus *instance* yang ada di dalam ontologi. Pengujian dilakukan dengan mencoba menghapus *instance* dengan nama “Kambing Guling” yang ada pada ontologi. Dengan mengklik tombol “Hapus” yang terdapat di

bawah nama dari *instance*, *instance* tersebut akan terhapus dari ontologi. Gambar 5.5, 5.6 dan tabel 5.5 akan menunjukkan dan menjabarkan lebih lanjut mengenai pengujian untuk fitur ini.



**Gambar 5.19 Hasil Pengujian Fitur Menghapus *Instance* dari Ontologi**



**Gambar 5.20 Hasil Pengujian Fitur Menghapus *Instance* dari Ontologi**

**Tabel 5.5 Pengujian Fitur Menghapus *Instance* dari Ontologi**

<b>No. Pengujian</b>	P-004
<b>Nama</b>	Pengujian fitur menghapus <i>instance</i> dari ontologi.
<b>Tujuan Pengujian</b>	Menguji fungsionalitas untuk fitur menghapus <i>instance</i> dari ontologi.
<b>Kondisi Awal</b>	<i>User</i> belum menghapus <i>instance</i> .
<b>Langkah Pengujian</b>	<i>User</i> memasukkan data mengenai <i>instance</i> yang ingin dimasukkan.
<b>Hasil yang diharapkan</b>	Sistem dapat menghapus <i>instance</i> dari ontologi.
<b>Hasil yang didapat</b>	<i>Instance</i> terkait sudah terhapus dari ontologi.
<b>Hasil Pengujian</b>	Berhasil
<b>Kondisi Akhir</b>	<i>Instance</i> terkait sudah tidak ada lagi di dalam ontologi.

### 5.3 Evaluasi Pengujian Fitur

Berdasarkan pengujian yang dilakukan, semua pengujian fitur memberikan hasil sesuai dengan scenario yang diharapkan. Hasil akhir mengenai pengujian berdasarkan fitur selanjutnya akan ditunjukkan pada tabel 5.6

**Tabel 5.6 Hasil Evaluasi Pengujian Fitur**

<b>No</b>	<b>Fitur</b>	<b>Terpenuhi</b>
1.	Melihat <i>Instance</i> dari Ontologi	✓
2.	Melihat Studi Kasus	✓
3.	Menambah <i>Instance</i> ke Dalam Ontologi	✓
3.	Menghapus <i>Instance</i> dari Ontologi	✓

### 5.4 Perbandingan Pengujian Fitur dengan Perangkat Bantu Protégé

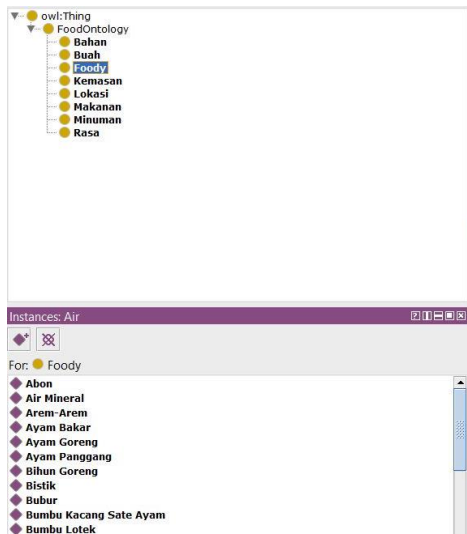
Pada subbab ini, hasil dari sebagian fitur yang diuji pada subbab sebelumnya, akan dibandingkan dengan menggunakan tools pembantu yaitu protégé. Protégé adalah sebuah perangkat lunak bantu yang digunakan untuk membuat, dan mengakses ontologi. Penjelasan lebih lengkap mengenai protégé dapat dilihat pada subbab 2.4.

Selain untuk meningkatkan keabsahan dari pengujian fitur, membandingkan pengujian dengan protégé bertujuan untuk mengecek apakah URI dari *instance* baru yang dimasukkan ke dalam ontologi, sudah memiliki format yang benar atau tidak.

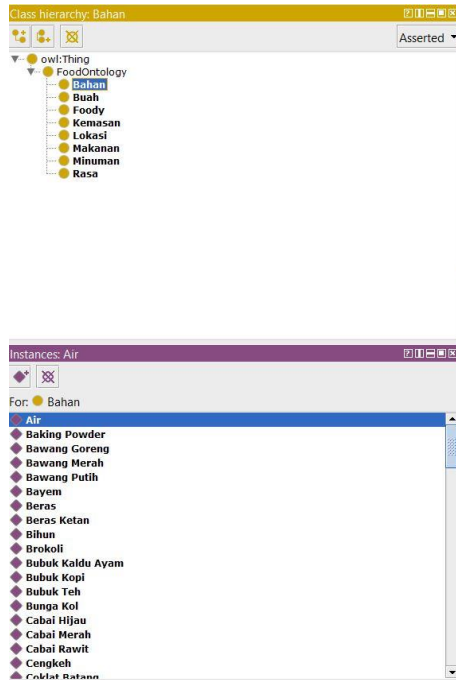
#### **5.4.1 Fitur Melihat *Instance* dari Ontologi**

Pada fitur ini, akan coba dibandingkan adalah apakah list dari *instance* juga dapat dilihat dari tools pembantu protégé, seperti layaknya list *instance* juga dapat dilihat dari sistem. Gambar 5.7 akan menunjukkan list *instance* dari kelas Foody sedangkan gambar 5.8 akan menunjukkan list *instance* dari kelas Bahan.





**Gambar 5.21** Hasil Pembacaan dari Protégé untuk fitur Melihat *Instance* dari Ontologi untuk kelas Foody



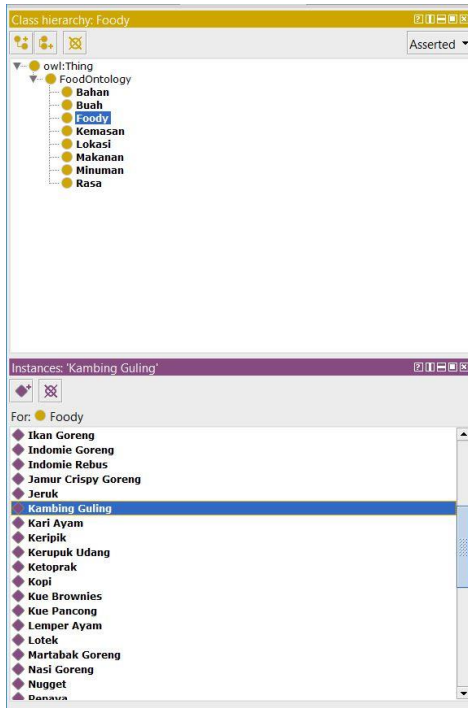
**Gambar 5.22 Hasil dari Pengujian pada Protégé untuk fitur Melihat Instance dari Ontologi untuk kelas Bahan**

Berdasarkan dari perbandingan diatas, maka kesimpulan pada fitur ini, *Output* nya sudah sesuai dengan *Output* menggunakan tools protégé.

## 5.4.2 Fitur Menambah Instance ke Dalam Ontologi

Pada fitur ini, yang akan dibandingkan adalah, apakah ketika *user* sudah menginput *instance* baru ke dalam ontologi melalui mekanisme di sistem, *instance* baru tersebut bisa dibaca di protégé. Gambar 5.10 akan menunjukkan hasil dari tools protégé ketika fitur menambah *instance* telah dijalankan oleh *user*. studi

kasus pada kali ini masih sama dengan pengujian sebelumnya, yaitu dengan menggunakan nama *instance* “Kambing Guling”



**Gambar 5.23 Hasil Pembacaan dari Protégé untuk Fitur Menambah *Instance* ke Dalam Ontologi**



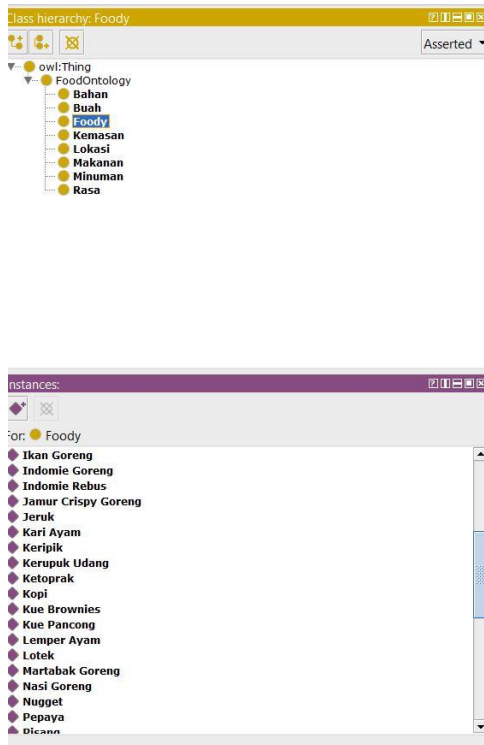
**Gambar 5.24 Hasil Pembacaan dari Protégé untuk Fitur Menambah *Instance* ke Dalam Ontologi**

Pada gambar 5.19, dapat dilihat URI untuk *instance* baru yang dimasukkan ke dalam ontologi, yaitu kambing guling. Pada URI, dapat dilihat terdapat penggunaan “%20” diantara kata “Kambing” dan “Guling”. Hal tersebut berarti bahwa terdapat spasi yang memisahkan antara kedua kata tersebut. Namun dikarenakan pada URI tidak diperbolehkan ada spasi, maka digunakan symbol “%20” yang menandakan ada spasi antara kata “Kambing” dan “Guling”.

Selanjutnya, dapat disimpulkan bahwa untuk fitur menambah *instance* ke dalam ontologi, untuk hasil *Output* dari sistem adalah sesuai dengan hasil *Output* dengan menggunakan protégé.

### 5.4.3 Fitur Menghapus *Instance* dari Ontologi

Pada fitur ini, yang akan dibandingkan adalah, apakah ketika *user* sudah memutuskan untuk menghapus suatu *instance*, maka *instance* tersebut masih terbaca di protégé atau memang sudah tidak dibaca lagi di protégé. Normal nya, ketika *user* sudah memutuskan untuk menghapus *instance*, maka *instance* tersebut seharusnya sudah tidak terbaca lagi di protégé. Gambar 5.11 akan menunjukkan pembacaan pada protégé ketika *user* sudah mengklik tombol hapus pada *instance*. Sama seperti pengujian, *instance* yang dipakai adalah *instance* dengan nama “Kambing Guling”



**Gambar 5.25 Hasil Pembacaan dari Protégé untuk Fitur Menghapus *Instance* dari Dalam Ontologi**

Selanjutnya, dapat disimpulkan bahwa untuk fitur *instance* dari dalam ontologi, untuk hasil *Output* dari sistem adalah sesuai dengan hasil *Output* dengan menggunakan protégé.

*[Halaman ini sengaja dikosongkan]*

## BAB VI

### KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

#### 6.1 Kesimpulan

Dari hasil pengamatan selama proses pengerjaan Tugas Akhir dari tahap perancangan, implementasi, hingga pengujian sistem, dapat diambil kesimpulan sebagai berikut

1. Studi kasus *food ontology* untuk aplikasi gifood.id dalam Bahasa Indonesia, mampu dimodelkan dengan mengembangkan ontologi dari nol. Pengembangan ontologi dari nol dilakukan dikarenakan tidak adanya ontologi yang sudah pernah dikembangkan spesifik untuk studi kasus tersebut.
2. *Library EasyRDF* mampu menghubungkan antara sistem dengan ontologi, lalu mampu menampilkan list beberapa *instance* yang ada, serta menampilkan informasi dari *instance* tersebut. *Library RDFlib* mampu untuk menguji ontologi dengan studi kasus yang sudah dikembangkan dengan SPARQL. Selain itu, *RDFlib* juga mampu mengakses ontologi, lalu melakukan manipulasi untuk menambah (*Create*) *instance* baru, dan menghapus (*Delete*) *instance* dari dalam ontologi.
3. Tugas akhir ini juga berhasil untuk mengembangkan sebuah sistem berbasis web dimana sistem tersebut dapat mengakses ontologi, dan melakukan manipulasi terhadap ontologi. Hal ini dapat dimungkinkan dengan

menggunakan beberapa *Library* yang khusus untuk menghandle file RDF dan mengakses ontologi.

4. Ontologi yang dikembangkan memiliki kurang lebih 8 kelas, 15 *object properties* (relasi), 3 *data properties* (atribut), dan 178 *instance* (individual).
5. Studi kasus dikembangkan menggunakan SPARQL untuk menguji kebenaran dari ontologi yang dikembangkan. Terdapat 10 studi kasus yang dikembangkan dan fungsionalitas dari studi kasus dapat diterapkan pada ontologi dengan memperoleh *Output* yang tepat.

## 6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini berdasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan.

1. Penambahan informasi lebih lanjut, contohnya mengenai kesehatan seperti penambahan kelas penyakit, sehingga nantinya ontologi bisa digunakan untuk mengidentifikasi makanan, minuman, atau buah apa yang sebaiknya tidak dikonsumsi untuk suatu penyakit tertentu.
2. Kedepan nya agar sistem yang dibuat dapat diintegrasikan langsung dengan sistem dari aplikasi gifood.id.



## DAFTAR PUSTAKA

- [1] H. Al-Zubaide and A. Issa, "OntBot : Ontology Based Chatbot," *Fourth International Symposium on Innovation in Information & Communication Technology*, 2011.
- [2] Snae and M. Brückner, "FOODS: A Food-Oriented Ontology-Driven System," *Department of Computer Science and Information Technology, Naresuan University, Phitsanulok, Thailand*, 2008.
- [3] Y. Badron, F. Agus and H. Hatta, "Studi Tentang Pemodelan Ontologi Web Semantik Dan Prospek Penerapan Pada Bibliografi Artikel Jurnal Ilmiah," *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, vol. 2, 2017.
- [4] O. Corcho, M. Fernandez-Lopez and A. Gomez-Perez, "Ontological Engineering : What Are Ontologies and How Can We Build Them?," *University of Manchester, UK*, 2007.
- [5] L. Liana, "Pemodelan Data," *Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Mercu Buana, Jakarta*, 2015.
- [6] T. Slimani, "Ontology Development: A Comparing Study on Tools, Languages and Formalisms," *Indian Journal of Science and Technology*, vol. 8, 2015.
- [7] W. W. W. Consortium, "What are OWL Ontologies?".
- [8] J. E. L. d. Vergara, V. A. Villagr a and J. Berrocal, "Applying the Web Ontology Language to management information definitions," *Telematic Systems Engineering Department, Technical University of Madrid*, 2003.
- [9] J. Perez, M. Arenas and C. Gutierrez, "Semantics and Complexity of SPARQL," *Universidad de Talca, Chile*, 2006.
- [10] M. M. Taye, "Web-Based Ontology Languages and its Based Description Logics," *The Research Bulletin of Jordan*, vol. 2, p. 9, 2011.
- [11] A. Mishra, "Critical Comparison Of PHP And ASP.NET For Web Development," *International Journal of Scientific & Technology Research*, vol. 3, 2014.
- [12] Z. T. Inc., "An overview on PHP," *Zend The PHP Company*, 2007.
- [13] K. R. Srinath, "Python – The Fastest Growing Programming Language," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 12, 2017.
- [14] N. Handayani, "MAKANAN KEMASAN : PROSES DAN PERMASALAHAN YANG DITIMBULKAN," *Redaksi Majalah Ilmiah Warta Dharmawangsa*, p. 20.

- [15] J. E. Clark, "Taste and flavour: their importance in food choice and acceptance," *Proceedings of the Nutrition Society*, 1998.

## LAMPIRAN A

<b>Instance pada kelas Bahan</b>	
<b>Nama Instance</b>	<b>URI</b>
Beras_ketan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Beras_Ketan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Beras_Ketan</a>
Beras	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Beras">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Beras</a>
Nasi	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nasi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nasi</a>
Daging_ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Ayam</a>
Daun Kemangi	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Kemangi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Kemangi</a>
Lengkuas	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lengkuas">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lengkuas</a>
Daun_Jeruk	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk</a>
Nangka	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nangka">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nangka</a>
Daun_Jati	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jati">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jati</a>
Daun_Salam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Salam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Salam</a>
Gula_Merah	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gula_Merah">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gula_Merah</a>
Kemiri	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kemiri">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kemiri</a>
Kaldu_Ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bubuk_Kaldu_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bubuk_Kaldu_Ayam</a>
Bawang_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Goreng</a>

Daun_Seledri	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Seledri">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Seledri</a>
Jeruk_Lemon	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jeruk_Lemon">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jeruk_Lemon</a>
Madu	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Madu">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Madu</a>
Lada/merica	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lada/merica">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lada/merica</a>
Daging_Kambing	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Kambing">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Kambing</a>
Ketumbar	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ketumbar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ketumbar</a>
Jeruk_Nipis	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jeruk_Nipis">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jeruk_Nipis</a>
Tepung_Beras	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Beras">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Beras</a>
Susu_Bubuk_Vanilla	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu_Bubuk_Vanilla">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu_Bubuk_Vanilla</a>
Coklat_Batang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Batang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Batang</a>
Coklat_Bubuk	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Bubuk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Bubuk</a>
Baking_Powder	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Baking_Powder">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Baking_Powder</a>
Tepung_Maizena	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Maizena">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Maizena</a>
Kuning_Telur	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kuning_Telur">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kuning_Telur</a>

Susu_Sapi	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu_Sapi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu_Sapi</a>
Tepung_Tapioka	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Tapioka">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Tapioka</a>
Bayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bayam</a>
Tauge	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tauge">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tauge</a>
Lontong	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lontong">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lontong</a>
Bunga_Kol	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bunga_Kol">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bunga_Kol</a>
Saus_Tiram	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Saus_Tiram">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Saus_Tiram</a>
Bihun	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bihun">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bihun</a>
Minyak_Sayur	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Minyak_Sayur">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Minyak_Sayur</a>
Tepung_Panir	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Panir">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Panir</a>
Kentang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kentang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kentang</a>
Wortel	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Wortel">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Wortel</a>
Mayonaise	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mayonaise">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mayonaise</a>
Terasi	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Terasi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Terasi</a>
Labu_Siam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Labu_Siam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Labu_Siam</a>
Nangka_Muda	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nangka_Muda">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nangka_Muda</a>
Terong	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Terong">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Terong</a>

Daun_Melinjo	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Melinjo">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Melinjo</a>
Telur_Bebek	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Telur_Bebek">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Telur_Bebek</a>
Daging_Sapi	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Sapi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daging_Sapi</a>
Jamur	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jamur">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jamur</a>
Indomie	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Indomie">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Indomie</a>
MSG	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#MSG">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#MSG</a>
Kacang_Panjang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tumis_Kacang_Panjang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tumis_Kacang_Panjang</a>
Laos	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Laos">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Laos</a>
Udang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang</a>
Tomat	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tomat">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tomat</a>
Tempe	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tempe">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tempe</a>
Tahu	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tahu">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tahu</a>
Kunyit	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kunyit">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kunyit</a>
Timun	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Timun">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Timun</a>
Kerupuk	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kerupuk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kerupuk</a>
Kacang_Tanah	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kacang_Tanah">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kacang_Tanah</a>
Kencur	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kencur">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kencur</a>

Gula_Merah	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gula_Merah">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gula_Merah</a>
Saus_Sambal	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Saus_Sambal">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Saus_Sambal</a>
Kecap_Ingggris	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecap_Ingggris">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecap_Ingggris</a>
Bubuk_Kaldu_Ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bubuk_Kaldu_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bubuk_Kaldu_Ayam</a>
Bunga_Kol	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bunga_Kol">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bunga_Kol</a>
Brokoli	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Brokoli">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Brokoli</a>
Sawi_Putih	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sawi_Putih">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sawi_Putih</a>
Royco	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Royco">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Royco</a>
Mie	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mie">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mie</a>
Bawang_Putih	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Putih">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Putih</a>
Bawang_Merah	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Merah">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bawang_Merah</a>
Lengkuas	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lengkuas">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lengkuas</a>
Cabai_Hijau	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cabai_Hijau">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cabai_Hijau</a>
Cabai_Merah	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cabai_Merah">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cabai_Merah</a>
Cabai_Rawit	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cabai_Rawit">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cabai_Rawit</a>

Kecap_Manis	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecap_Manis">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecap_Manis</a>
Kecap_Asin	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecap_Asin">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecap_Asin</a>
Gula_Pasir	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gula_Pasir">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gula_Pasir</a>
Jinten	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jinten">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jinten</a>
Garam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Garam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Garam</a>
Tepung_Terigu	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Terigu">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tepung_Terigu</a>
Coklat_Bubuk	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Bubuk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Bubuk</a>
Coklat_Batang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Batang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Coklat_Batang</a>
Air	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Air">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Air</a>
Keju	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Keju">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Keju</a>
Daun_Bawang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Bawang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Bawang</a>
Daun_Salam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Salam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Salam</a>
Serai	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Serai">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Serai</a>
Wijen	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Wijen">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Wijen</a>
Kayu_Manis	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kayu_Manis">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kayu_Manis</a>
Kapulaga	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kapulaga">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kapulaga</a>



Jahe	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jahe">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jahe</a>
Cengkeh	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cengkeh">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Cengkeh</a>
Minyak_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Minyak_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Minyak_Goreng</a>
Telur_Ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Telur_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Telur_Ayam</a>
Santan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Santan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Santan</a>
Mentega	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mentega">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mentega</a>

**LAMPIRAN B**

<b>Instance pada kelas Buah</b>	
<b>Nama Instance</b>	<b>URI</b>
Jeruk	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk</a>
Pepaya	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pepaya">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pepaya</a>
Pisang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Pisang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Pisang</a>
Salak	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Salak">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Salak</a>
Semangka	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Semangka">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Semangka</a>

**LAMPIRAN C**

<b>Instance pada kelas Buah</b>	
<b>Nama Instance</b>	<b>URI</b>
Jeruk	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Jeruk</a>
Pepaya	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pepaya">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pepaya</a>
Pisang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Pisang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Pisang</a>
Salak	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Salak">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Salak</a>
Semangka	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Semangka">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Semangka</a>

## LAMPIRAN D

<b>Instance pada kelas Kemasan</b>	
<b>Nama Instance</b>	<b>URI</b>
Aluminium_Foil	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Aluminium_Foil">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Aluminium_Foil</a>
Botol_Kaca	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Botol_Kaca">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Botol_Kaca</a>
Botol_Plastik	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Botol_Plastik">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Botol_Plastik</a>
Daun_Pisang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Pisang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Daun_Pisang</a>
Gelas_Plastik	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gelas_Plastik">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gelas_Plastik</a>
Kertas_Minyak_Makanan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kertas_Minyak_Makanan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kertas_Minyak_Makanan</a>
Kotak	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kotak">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kotak</a>
Mika	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mika">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Mika</a>
Piring	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Piring">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Piring</a>
Plastik	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Plastik">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Plastik</a>
Plastik_Kemasan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Plastik_Kemasan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Plastik_Kemasan</a>
Sterofoam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sterofoam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sterofoam</a>

**LAMPIRAN E**

<b>Instance pada kelas Lokasi</b>	
<b>Nama Instance</b>	<b>URI</b>
Jalan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jalan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jalan</a>
Kabupaten	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kabupaten">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kabupaten</a>
Kecamatan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecamatan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kecamatan</a>
Kelurahan	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kelurahan">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kelurahan</a>

## LAMPIRAN F

<i>Instance pada kelas Makanan</i>	
<b>Nama Instance</b>	<b>URI</b>
Nasi_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nasi_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Nasi_Goreng</a>
Gudeg	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gudeg">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Gudeg</a>
Bubur_Ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bubur_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bubur_Ayam</a>
Ayam_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ayam_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ayam_Goreng</a>
Ayam_Bakar	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ayam_Bakar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ayam_Bakar</a>
Ayam_Panggang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ayam_Panggang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ayam_Panggang</a>
Sate_Ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sate_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sate_Ayam</a>
Sate_Kambing	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sate_Kambing">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sate_Kambing</a>
Kue_Pancong	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kue_Pancong">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kue_Pancong</a>
Kue_Brownies	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kue_Brownies">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kue_Brownies</a>
Risoles	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Risoles">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Risoles</a>
Sambal	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sambal">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sambal</a>

Sayur_Lodeh	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sayur_Lodeh">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sayur_Lodeh</a>
Martabak_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Martabak_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Martabak_Goreng</a>
Jamur_crispy_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jamur_Crispy_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Jamur_Crispy_Goreng</a>
Indomie_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Indomie_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Indomie_Goreng</a>
Indomie_Rebus	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Indomie_Rebus">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Indomie_Rebus</a>
Tumis_Kacang_Panjang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tumis_Kacang_Panjang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tumis_Kacang_Panjang</a>
Udang_Bakar	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang_Bakar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang_Bakar</a>
Kerupuk_Udang	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kerupuk_Udang">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kerupuk_Udang</a>
Udang_Balado	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang_Balado">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang_Balado</a>
Udang_Bakar	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang_Bakar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Udang_Bakar</a>
Telur_Dadar	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Telur_Dadar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Telur_Dadar</a>
Tahu_Bakso	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tahu_Bakso">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tahu_Bakso</a>

Tahu_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tahu_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tahu_Goreng</a>
Tempe_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tempe_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tempe_Goreng</a>
Sayur_Tempe_Tahu	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sayur_Tempe_Tahu">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sayur_Tempe_Tahu</a>
Lotek	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lotek">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lotek</a>
Bumbu_Lotek	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bumbu_Lotek">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bumbu_Lotek</a>
Bistik	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bistik">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bistik</a>
Capcay	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Capcay">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Capcay</a>
Bumbu_Kacang_Sate_Ayam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bumbu_Kacang_Sate_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bumbu_Kacang_Sate_Ayam</a>
Ikan_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ikan_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Ikan_Goreng</a>
Arem_Arem	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Arem_Arem">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Arem_Arem</a>
Tumis_Tempe	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tumis_Tempe">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tumis_Tempe</a>
Bihun_Goreng	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bihun_Goreng">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Bihun_Goreng</a>
Abon	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Abon">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Abon</a>
Lemper	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lemper">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Lemper</a>



Kari_Ayam	<i><a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kari_Ayam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kari_Ayam</a></i>
-----------	--

**LAMPIRAN G**

<b><i>Instance pada kelas Minuman</i></b>	
<b><i>Nama Instance</i></b>	<b><i>URI</i></b>
Air_Mineral	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Air_Mineral">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Air_Mineral</a>
Kopi	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kopi">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Kopi</a>
Sirup	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sirup">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Sirup</a>
Susu	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Susu</a>
Teh	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Teh">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Teh</a>

**LAMPIRAN H**

<b><i>Instance pada kelas Rasa</i></b>	
<b><i>Nama Instance</i></b>	<b><i>URI</i></b>
Asam	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Asam">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Asam</a>
Asin	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Asin">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Asin</a>
Manis	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Manis">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Manis</a>
Pahit	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pahit">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pahit</a>
Pedas	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pedas">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Pedas</a>
Tawar	<a href="http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tawar">http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#Tawar</a>

## LAMPIRAN I

### Implementasi Fitur Menambah Instance ke Dalam Ontologi

```
1. from rdflib import Graph
2. from rdflib.namespace import RDF, FOAF, RDFS, OWL
3. from rdflib import URIRef, BNode, Literal
4. from rdflib.namespace import XSD
5. import rdflib
6. import sys
7. import json
8.
9. kelas = eval(sys.argv[1])[0]
10. individu = eval(sys.argv[1])[1]
11.
12. RDF.type
13. RDF.datatype
14. RDFS.label
15. RDFS.comment
16. RDFS.seeAlso
17. OWL.About
18.
19. FOAF.knows
20.
21. g = Graph()
22. g.parse("C:\\xampp\\htdocs\\foody\\foody.rdf")
23.
24. var_kelas = URIRef("http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#" + kelas)
25. var_individu = URIRef("http://www.semanticweb.org/asus/ontologies/2019/1/untitled-ontology-49#" + individu)
26.
27. g.add( (var_individu, RDF.type, var_kelas ) )
28. g.add( (var_individu, RDFS.label, Literal(individu, lang="en")) )
29.
30. g.serialize("C:\\xampp\\htdocs\\foody\\foody.rdf")
```

## LAMPIRAN J

### Implementasi Fitur Menghapus Instance dari Ontologi

```
1. from rdflib import Graph
2. from rdflib.namespace import RDF, FOAF, RDFS, OWL
3. from rdflib import URIRef, BNode, Literal
4. from rdflib.namespace import XSD
5. import rdflib
6. import sys
7. import json
8.
9. entity = eval(sys.argv[1])[0]
10.
11. RDF.type
12. RDF.datatype
13. RDFS.label
14. RDFS.comment
15. RDFS.seeAlso
16. OWL.About
17.
18. g = Graph()
19. g.parse("C:\\xampp\\htdocs\\foody\\foody.rdf")
20.
21. entiti = URIRef("http://www.semanticweb.org/asus/
    ontologies/2019/1/untitled-ontology-
    49#+entity")
22.
23. g.remove( (entiti, None, None) )
24.
25. g.serialize("C:\\xampp\\htdocs\\foody\\foody.rdf"
    )
```

## LAMPIRAN K

### Skenario Hasil Pengujian Studi Kasus untuk *True Positive* dan *True Negative*

<b>Pengujian Studi Kasus</b>			
<b>No.</b>	<b>Studi Kasus</b>	<b>Dugaan</b>	<b>Hasil Pengujian</b>
1	Menampilkan Bahan yang dipakai untuk Makanan dengan cara penyajian di Rebus	Benar	Benar
2	Menampilkan Makanan yang dikemas menggunakan Daun Pisang	Benar	Benar
3	Menampilkan kandungan bahan pada makanan capcay	Benar	Benar
4	Menampilkan Minuman yang Memiliki Rasa Pahit	Benar	Benar
5	Menampilkan Bahan yang bila dipanggang akan menghasilkan makanan yang memiliki rasa gurih	Benar	Benar
6	Menampilkan Makanan yang Memiliki Rasa Pedas	Benar	Benar
7	Menampilkan Bahan yang digunakan sebagai pelengkap dari suatu makanan beserta nama makanan terkait	Benar	Benar
8	Menampilkan Makanan yang Dikemas Dengan Plastik	Benar	Benar
9	Menampilkan Kemasan yang digunakan untuk sate ayam	Benar	Benar
10	Menampilkan Minuman yang Memiliki Rasa Manis	Benar	Salah
S1	Menampilkan Bahan yang digunakan sebagai pelengkap dari suatu makanan beserta nama makanannya	Salah	Salah

S2	Menampilkan Kandungan Bahan pada Makanan Capcay	Salah	Salah
S3	Menampilkan Makanan untuk Porsi Individu	Salah	Salah
S4	Menampilkan Bahan yang Dipakai untuk Makanan dengan Penyajian di Rebus	Salah	Salah
S5	Menampilkan Makanan yang Memiliki Rasa yang Pedas	Salah	Salah

## BIODATA PENULIS



Wahyu Ivan Satyagraha, lahir di Jakarta pada tanggal 17 Maret 1997. Penulis telah menempuh pendidikan formal mulai dari SD Islam Al-Azhar 9 Bekasi, SMP Islam Al-Azhar 9 Bekasi, SMA Labschool Jakarta dan terakhir sebagai mahasiswa Departemen Informatika Institut Teknologi Sepuluh Nopember dengan rumpun mata kuliah Manajemen Informasi (2015-2019). Lulus dari SMA, penulis melanjutkan pendidikan di Departemen Informatika.

Selama perkuliahan, penulis aktif dalam organisasi kemahasiswaan, antara lain sebagai Staff Logistik FTIf Festival 2016, Staff Departemen Dalam Negeri Badan Eksekutif Mahasiswa Fakultas (BEMf) Fakultas Teknologi dan Informasi (FTIF) ITS 2016-2017, Staff Ahli Acara FTIf Festival 2017, Kepala Divisi Eksternal Departemen Hubungan Luar Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS 2017-2018, Hingga menjadi Koordinator Sub-event National Seminar of Technology (NST) Schematics 2017.

Selama kuliah di Departemen Informatika ITS, penulis mengambil bidang minat Manajemen Informasi (MI) dengan ketertarikan penulis pada bagaimana mendapatkan, mengolah, dan menyampaikan data dengan baik dan benar. Penulis dapat dihubungi melalui surel [ivan.satyagraha@gmail.com](mailto:ivan.satyagraha@gmail.com).



