



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IF184802

***VIRTUAL ASSISTANT CHATBOT PADA APLIKASI
GIFOOD.ID MENGGUNAKAN SPEECH
RECOGNITION DENGAN NATURAL LANGUAGE
PROCESSING***

SATRIYO NUGROHO
NRP 0511154000034

Dosen Pembimbing
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Kelly Rossa Sungkono, S.Kom., M.Kom.

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



TUGAS AKHIR - IF184802

***VIRTUAL ASSISTANT CHATBOT PADA APLIKASI
GIFOOD.ID MENGGUNAKAN SPEECH
RECOGNITION DENGAN NATURAL LANGUAGE
PROCESSING***

**SATRIYO NUGROHO
NRP 0511154000034**

**Dosen Pembimbing
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Kelly Rossa Sungkono, S.Kom., M.Kom.**

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

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - IF184802

***VIRTUAL ASSISTANT CHATBOT IN GIFOOD.ID
APPLICATIONS USING SPEECH RECOGNITION
WITH NATURAL LANGUAGE PROCESSING***

SATRIYO NUGROHO
NRP 0511154000034

Supervisors
Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Kelly Rossa Sungkono, S.Kom., M.Kom.

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

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

VIRTUAL ASSISTANT CHATBOT PADA APLIKASI GIFOOD.ID MENGGUNAKAN SPEECH RECOGNITION DENGAN NATURAL LANGUAGE PROCESSING

TUGAS AKHIR

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

Oleh:

SATRIYO NUGROHO

NRP : 0511154000034

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
NIP: 19650518 199203 1 003
(Pembimbing 1)
2. Kelly Rossa Sungkono, S.Kom.,
M.Kom.
NIP: 199420 191208 8
(Pembimbing 2)

**SURABAYA
JULI 2019**

[Halaman ini sengaja dikosongkan]

VIRTUAL ASSISTANT CHATBOT PADA APLIKASI GIFOOD.ID MENGGUNAKAN SPEECH RECOGNITION DENGAN NATURAL LANGUAGE PROCESSING

Nama Mahasiswa : Satriyo Nugroho
NRP : 0511154000034
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Kelly Rossa Sungkono, S.Kom.,
M.Kom.

ABSTRAK

Chatbot merupakan program komputer yang meniru percakapan cerdas. Chatbot dapat membantu dalam interaksi komputer manusia dan mereka memiliki kemampuan untuk memeriksa dan mempengaruhi perilaku pengguna dengan mengajukan pertanyaan dan menanggapi pertanyaan-pertanyaan pengguna. Hingga saat ini, banyak perusahaan yang menggunakan chatbot untuk menambah layanan mereka kepada konsumen.

Pada tugas akhir ini akan dibangun chatbot untuk salah satu start-up yang bergerak untuk berbagi makanan yang berlebih kepada orang yang membutuhkan, yaitu Gifood.id. Chatbot ini dibangun pada platform web menggunakan Natural Language Processing (NLP) untuk mengubah dari bahasa alami menjadi query, ontologi untuk menyimpan data, dan ditambahkan speech recognition sebagai salah satu fitur untuk konsumen agar lebih interaktif dengan chatbot. Natural Language Processing (NLP) sendiri merupakan salah satu cabang ilmu Artificial Intelligence (AI) yang berfokus pada pengolahan bahasa natural.

Chatbot dibangun untuk membantu konsumen untuk mengetahui apa saja yang ada pada Gifood, melakukan transaksi

berbagi makanan, dan mencari makanan sesuai yang diinginkan. Chatbot yang telah dibangun layak dan diterima oleh masyarakat menurut System Usability Scale (SUS).

Dengan adanya chatbot ini, konsumen dapat lebih efisien, mudah, dan praktis dalam menggunakan layanan dari gifood.id dan juga interaktif dengan sistem Gifood layaknya berinteraksi dengan manusia.

Kata kunci: Chatbot, Natural Language Processing, Ontologi, Speech Recognition

VIRTUAL ASSISTANT CHATBOT IN GIFOOD.ID APPLICATIONS USING SPEECH RECOGNITION WITH NATURAL LANGUAGE PROCESSING

Name : Satriyo Nugroho
NRP : 05111540000034
Major : Informatics Department FTIK-ITS
Supervisor I : Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Supervisor II : Kelly Rossa Sungkono, S.Kom.,
M.Kom.

ABSTRACT

A chatbot is a computer program that mimics intelligent conversation. A chatbot can help in human-computer interactions, and it can examine and influence user behaviour by asking questions and responding to user questions. Until now, many companies use chatbots to add their services to consumers.

In this final assignment, a chatbot will be built for one of the start-up that moves to share excess food with people in need, namely Gifood.id. The chatbot is built on a web platform using Natural Language Processing (NLP) for changing natural language to query, ontology for storing data, and speech recognition as a feature for giving consumers more interactive environment. Natural Language Processing (NLP) is one branch of Artificial Intelligence (AI) that focuses on natural language.

Chabot was built to help consumers to know what Gifood is, do food sharing transactions, and find food as desired. Chatbot that has been built is feasible and accepted by people according to the System Usability Scale (SUS).

With this chatbot, consumers can be more efficient, easy and practical in using the services of Gifood.id and also interactive with the Gifood system to interact with humans.

***Keywords: Chatbot, Natural Language Processing,
Ontology, Speech Recognition***

KATA PENGANTAR

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

VIRTUAL ASSISTANT CHATBOT PADA APLIKASI GIFOOD.ID MENGGUNAKAN SPEECH RECOGNITION DENGAN NATURAL LANGUAGE PROCESSING

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 Sujiman dan Ibu Sri Astini, kakak Liila Masru’ah, kakak Pambudi Santoso, kakak Nur Hidayati dan kakak Fahmi Pratama yang telah memberikan doa dan dukungan selama ini.
3. Bapak Dr.tech. Ir. R.V.Hari Ginardi, M.Sc. selaku dosen pembimbing I yang selalu memberikan masukan, arahan, motivasi dan membimbing penulis selama pengerjaan Tugas Akhir.
4. Ibu Kelly Rossa Sungkono, S.Kom., M.Kom. selaku dosen pembimbing II yang senantiasa memberikan masukan, arahan, dan bantuan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
5. Bapak dan Ibu dosen Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Informatika ITS.
6. Vebriane Werdaningrum yang telah menemani dan memberikan dukungan penulis sampai saat ini.
7. Gifood.id yang telah membantu dan memberikan data dalam

pembuatan Tugas Akhir ini.

8. Teman seperjuangan Tugas Akhir; Ivan Satyagraha yang saling membantu, menyemangati, dan saling koordinasi selama mengerjakan Tugas Akhir di Laboratorium Manajemen Informasi.
9. Sahabat Gaes, Ivan Fadhila, Yuga, Abyan, Zahri, Adib, Narendra, Rafi, dan Wahyu yang selalu memberi warna kehidupan penulis selama kuliah di Informatika ITS.
10. Teman-teman Pengurus Harian Schematics HMTc ITS 2017 yang telah memberikan kesempatan penulis untuk mendapatkan pengalaman lebih selama 1 tahun kepengurusan.
11. Teman-teman Administrator Laboratorium MI yang menjadi keluarga selama penulis menimba ilmu di Informatika ITS.
12. Teman-teman angkatan 2015 yang telah memberikan motivasi selama penulis berkuliah di Informatika ITS.
13. 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

Satriyo Nugroho

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR SINGKATAN	xxi
DAFTAR KODE SUMBER	xxiii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi Pembuatan Tugas Akhir.....	3
1.7 Sistematika Penulisan	6
2 BAB II TINJAUAN PUSTAKA.....	9
2.1 <i>Literature Review</i> Penelitian Terdahulu	9
2.2 <i>Chatbot</i>	11
2.3 <i>Natural Language Processing</i>	12
2.3.1 Tokensisasi	13
2.3.2 <i>Lemma</i>	14
2.3.3 <i>POS Tag</i>	14
2.4 <i>Speech Recognition</i>	14
2.5 <i>Natural Language Toolkit</i>	15
2.6 <i>Quepy</i>	16
2.7 <i>Ontologi</i>	16
2.8 <i>RDF</i>	17
2.9 <i>RDFlib</i>	18
3 BAB III ANALISIS DAN PERANCANGAN SISTEM....	19
3.1 Analisis Metode Secara Umum.....	19

3.1.1 Analisis Permasalahan	19
3.1.2 Deskripsi Umum Sistem	20
3.1.3 Spesifikasi Kebutuhan Perangkat Lunak.....	23
3.2 Perancangan	28
3.2.1 Perancangan Arsitektur Sistem.....	28
3.2.2 Perancangan Ontologi	29
3.2.3 Perancangan Antarmuka Pengguna	30
3.2.4 Diagram Alur Sistem	33
4 BAB IV IMPLEMENTASI.....	37
4.1 Lingkungan Implementasi Perangkat Lunak	37
4.2 Implementasi Antarmuka Pengguna.....	37
4.2.1 Implementasi Halaman <i>Chatbot</i>	38
4.2.2 Implementasi Halaman Berbagi Makanan	39
4.3 Implementasi Fungsi Objek, Predikat, <i>Prefix</i>	40
4.3.1 Implementasi Fungsi Objek.....	40
4.3.2 Implementasi Fungsi Predikat	41
4.3.3 Implementasi Fungsi Prefix.....	41
4.4 Implementasi <i>Speech Recognition</i>	41
4.5 Implementasi Kasus Penggunaan.....	43
4.5.1 Implementasi Kasus Penggunaan Jenis Makanan yang dapat Dibagikan	43
4.5.2 Implementasi Kasus Penggunaan Apa itu Gifood	44
4.5.3 Implementasi Kasus Penggunaan Cara Kerja Gifood	44
4.5.4 Implementasi Kasus Penggunaan Contact Person Gifood	45
4.5.5 Implementasi Kasus Penggunaan Mitra Gifood	45
4.5.6 Implementasi Kasus Penggunaan Cara Mengetahui Kelayakan Makanan	46
4.5.7 Implementasi Kasus Penggunaan Apa itu Food Warrior	47
4.5.8 Implementasi Kasus Penggunaan Cara Menjadi Food Warrior	47
4.5.9 Implementasi Kasus Penggunaan Mencari Volunteer	48
4.5.10 Implementasi Kasus Penggunaan Makanan Berlebih	48
4.5.11 Implementasi Kasus Penggunaan Mencari Makanan yang Tersedia	49

4.5.12 Implementasi Kasus Penggunaan Berbagi makanan Berlebih	56
4.6 Implementasi Proses Mengubah Bahasa Alami Menjadi Query SPARQL.....	57
4.6.1 Implementasi Fungsi Main	57
4.6.2 Implementasi Fungsi Tokenisasi dan POS Tagging	59
4.6.3 Implementasi Fungsi Mengubah Menjadi Query SPAQRL	59
5 BAB V PENGUJIAN DAN EVALUASI	61
5.1 Lingkungan Pengujian	61
5.2 Pengujian <i>Speech Recognition</i>	61
5.3 Pengujian Fungsionalitas	62
5.3.1 Pengujian Fungsionalitas Jenis Makanan yang Dapat di Bagikan.	63
5.3.2 Pengujian Fungsionalitas Apa itu Gifood.....	64
5.3.3 Pengujian Fungsionalitas Cara Kerja Gifood	66
5.3.4 Pengujian Fungsionalitas Contact Person Gifood.....	67
5.3.5 Pengujian Fungsionalitas Mitra Gifood.	69
5.3.6 Pengujian Fungsionalitas Cara Mengetahui Kelayakan Makanan.....	70
5.3.7 Pengujian Fungsionalitas Apa itu Food Warrior.....	72
5.3.8 Pengujian Fungsionalitas Cara Menjadi Food Warrior.	74
5.3.9 Pengujian Fungsionalitas Mencari Volunteer.....	76
5.3.10 Pengujian Fungsionalitas Makanan Berlebih.	77
5.3.11 Pengujian Fungsionalitas Mencari Makanan yang Tersedia dengan Nama, Lokasi, Waktu, Status Ketersediaan, dan Jumlah Porsi.	79
5.3.12 Pengujian Fungsionalitas Berbagi Makanan Berlebih.	88
5.4 Pengujian <i>Natural Language Processing</i>	92
5.5 Evaluasi Pengujian Fungsionalitas	94
5.6 Hasil Ujicoba Pengguna	95
6 BAB VI KESIMPULAN DAN SARAN	97
6.1 Kesimpulan	97
6.2 Saran.....	98
DAFTAR PUSTAKA	99

LAMPIRAN A	101
LAMPIRAN B	103
LAMPIRAN C	107
LAMPIRAN D	109
LAMPIRAN E.....	111
INDEX	115
DAFTAR ISTILAH.....	117
BIODATA PENULIS	119

DAFTAR GAMBAR

Gambar 3.1 Deskripsi Umum Sistem	20
Gambar 3.2 Tahapan pada <i>Natural Language Processing</i>	21
Gambar 3.3 Diagram Kasus Penggunaan.....	27
Gambar 3.4 Arsitektur Sistem <i>Chatbot</i>	28
Gambar 3.5 <i>Conceptual Data Model</i> Pada <i>Database Plugin Meaningful Learning</i>	29
Gambar 3.6 Rancangan Halaman Antarmuka Chatbot	31
Gambar 3.7 Rancangan Halaman Antarmuka Form Membagikan Makanan.....	32
Gambar 3.8 Diagram Alur Sistem	35
Gambar 4.1 Implementasi Halaman Chatbot	38
Gambar 4.2 Implementasi Halaman Berbagi Makanan	39
Gambar 5.1 Ketika Tombol Microphone Ditekan	62
Gambar 5.2 Hasil Pengujian <i>Speech Recognition</i>	62
Gambar 5.3 Hasil Pengujian Fungsionalitas Jenis Makanan yang Dapat di Bagikan.	64
Gambar 5.4 Hasil Pengujian Fungsionalitas Apa itu Gifood.	65
Gambar 5.5 Hasil Pengujian Fungsionalitas Cara Kerja Gifood.	67
Gambar 5.6 Hasil Pengujian Fungsionalitas Contact Person Gifood.	68
Gambar 5.7 Hasil Pengujian Fungsionalitas Mitra Gifood.	70
Gambar 5.8 Hasil Pengujian Fungsionalitas Cara Mengetahui Kelayakan Makanan.....	72
Gambar 5.9 Hasil Pengujian Fungsionalitas Apa itu Food Warrior.	74
Gambar 5.10 Hasil Pengujian Fungsionalitas Cara Menjadi Food Warrior.....	75
Gambar 5.11 Hasil Pengujian Fungsionalitas Mencari Volunteer.	77
Gambar 5.12 Hasil Pengujian Fungsionalitas Makanan Berlebih.	78
Gambar 5.13 Hasil Pengujian Fungsionalitas Mencari Semua Makanan yang Tersedia	80

Gambar 5.14 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Nama Makanan.....	82
Gambar 5.15 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Jumlah Porsi	84
Gambar 5.16 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Tanggal	86
Gambar 5.17 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Lokasi	88
Gambar 5.18 Pengisian Form Berbagi Makanan.....	90
Gambar 5.19 Notifikasi Bahwa Makanan Yang Diinginkan Berhasil	91
Gambar 5.20 Hasil Pengujian Fungsionalitas Berbagi Makanan Berlebih.....	91
Gambar 5.21 Hasil Pengujian dengan Satu Kata.....	93
Gambar 5.22 Hasil Pengujian <i>Natural Language Processing</i>	94
Gambar 5.23 Skala Perhitungan SUS	96

DAFTAR TABEL

Tabel 2.1 <i>Literature Review</i> Penelitian Terdahulu	9
Tabel 3.1 Contoh Hasil Tokenisasi	21
Tabel 3.2 Contoh Hasil Pos Tag	22
Tabel 3.3 Contoh Proses Perubahan dari Bahasa Alami Menjadi Query Sparql	23
Tabel 3.4 Kebutuhan Fungsional	23
Tabel 3.5 Kualitas Perangkat Lunak	25
Tabel 3.6 Kasus Penggunaan	25
Tabel 3.7 Atribut Antarmuka Form Membagikan Makanan	33
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak	37
Tabel 4.2 Penjelasan Kode Sumber 9.1	57
Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak	61
Tabel 5.2 Pengujian <i>Speech Recognition</i>	62
Tabel 5.3 Pengujian Fungsionalitas Jenis Makanan yang Dapat di Bagikan.	63
Tabel 5.4 Pengujian Fungsionalitas Apa itu Gifood.	64
Tabel 5.5 Pengujian Fungsionalitas Cara Kerja Gifood	66
Tabel 5.6 Pengujian Fungsionalitas Contact Person Gifood.	67
Tabel 5.7 Pengujian Fungsionalitas Mitra Gifood.	69
Tabel 5.8 Pengujian Fungsionalitas Cara Mengetahui Kelayakan Makanan.	71
Tabel 5.9 Pengujian Fungsionalitas Apa itu Food Warrior.	73
Tabel 5.10 Pengujian Fungsionalitas Cara Menjadi Food Warrior.	74
Tabel 5.11 Pengujian Fungsionalitas Mencari Volunteer.	76
Tabel 5.12 Pengujian Fungsionalitas Makanan Berlebih.	77
Tabel 5.13 Pengujian Fungsionalitas Mencari Semua Makanan yang Tersedia	79
Tabel 5.14 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Nama Makanan	80
Tabel 5.15 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Jumlah Porsi	83

Tabel 5.16 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Tanggal	85
Tabel 5.17 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Lokasi	87
Tabel 5.18 Pengujian Fungsionalitas Berbagi Makanan Berlebih.	89
Tabel 5.19 Evaluasi Pengujian Fungsionalitas	94
Tabel 5.20 Hasil Ujicoba Pengguna	96
Tabel 9.1 Penjelasan Kode Sumber 9.1	106
Tabel 10.1 Tabel Part-of-Speech	107
Tabel 11.1 Kuesioner Pengujian Terhadap Chatbot	109
Tabel 12.1 Tabel Skenario dan Hasil Pengujian <i>True Positive</i> untuk <i>Natural Language Processing</i>	111
Tabel 12.2 Tabel Skenario dan Hasil Pengujian <i>True Negative</i> untuk <i>Natural Language Processing</i>	113

DAFTAR SINGKATAN

NLP	:	<i>Natural Language Processing</i>
NLTK	:	<i>Natural Language Toolkit</i>
AI	:	<i>Artificial Intelligence</i>
ML	:	<i>Machine Learning</i>
HTML	:	<i>Hypertext Markup Language</i>
PHP	:	<i>Hypertext Preprocessor</i>
RDF	:	<i>Resource Description Framework</i>
MQML	:	<i>MetaQuotes Language</i>
XML	:	<i>eXtensible Markup Language</i>
RDFS	:	<i>Resource Description Framework Schema</i>
URI	:	<i>Uniform Resource Identifier</i>
OWL	:	<i>Web Ontology Language</i>
POS	:	<i>Part of Speech</i>
UC	:	<i>Use Case</i>
SUS	:	<i>System Usability Scale</i>

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Fungsi Thing	40
Kode Sumber 4.2 Implementasi Fungsi Quantity	40
Kode Sumber 4.3 Implementasi Fungsi Date	41
Kode Sumber 4.4 Implementasi Fungsi Prefix	41
Kode Sumber 4.5 Implementasi Speech Recognition	42
Kode Sumber 4.6 Implementasi Kasus Penggunaan Jenis Makanan yang dapat Dibagikan	43
Kode Sumber 4.7 Implementasi Kasus Penggunaan Apa itu Gifood	44
Kode Sumber 4.8 Implementasi Kasus Penggunaan Cara Kerja Gifood	45
Kode Sumber 4.9 Implementasi Kasus Penggunaan Contact Person Gifood	45
Kode Sumber 4.10 Implementasi Kasus Penggunaan Mitra Gifood	46
Kode Sumber 4.11 Implementasi Kasus Penggunaan Cara Mengetahui Kelayakan Makanan	46
Kode Sumber 4.12 Implementasi Kasus Penggunaan Apa itu Food Warrior	47
Kode Sumber 4.13 Implementasi Kasus Penggunaan Cara Menjadi Food Warrior	47
Kode Sumber 4.14 Implementasi Kasus Penggunaan Mencari Volunteer	48
Kode Sumber 4.15 Implementasi Kasus Penggunaan Makanan Berlebih	49
Kode Sumber 4.16 Implementasi Pencarian Semua Makanan yang Tersedia	50
Kode Sumber 4.17 Implementasi Pencarian Makanan Berdasarkan Nama Makanan	51
Kode Sumber 4.18 Implementasi Pencarian Makanan Berdasarkan Jumlah Porsi	53
Kode Sumber 4.19 Implementasi Pencarian Makanan Berdasarkan Kota	54

Kode Sumber 4.20 Implementasi Pencarian Makanan Berdasarkan Tanggal	55
Kode Sumber 4.21 Implementasi Menambahkan Nama Makanan Baru	57
Kode Sumber 4.22 Implementasi Fungsi Membaca File RDF....	58
Kode Sumber 4.23 Implementasi Mengubah Bahasa Indonesia Menjadi Bahasa Inggris	58
Kode Sumber 4.24 Implementasi Fungsi Query Sparql.....	59
Kode Sumber 4.25 Implementasi Fungsi Tokenisasi dan POS Tagging	59
Kode Sumber 4.26 Implementasi Fungsi Mengubah Menjadi Query SPAQRL.....	60
Kode Sumber 8.1 Implementasi Predikat.....	102
Kode Sumber 9.1 Implementasi Kasus Penggunaan Berbagai Makanan Berlebih.....	105

BAB I

PENDAHULUAN

Bab ini akan memaparkan 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

Pada jaman sekarang *Artificial Intelligence* atau yang biasa disebut AI mulai banyak dimanfaatkan secara nyata dalam kehidupan sehari - hari, salah satu yang paling populer saat ini adalah *chatbot*. Bahkan saat ini *chatbot* sendiri banyak digunakan oleh banyak perusahaan. Umumnya *chatbot* dipadukan dengan *Natural Language Processing* (NLP) dan *Machine Learning* (ML) untuk mampu memahami dan menanggapi *request* dari para pengguna.

Hingga saat ini lebih dari 100.000 *chatbot* dapat kita temukan di berbagai aplikasi perpesanan [1]. Munculnya teknologi baru selalu mengubah kebiasaan konsumen serta mempengaruhi pola konsumsi dan interaksi konsumen dengan *brand*. Salah satu teknologi terbaru yang kini banyak dibahas adalah *chatbot*. Menurut survei Oracle dengan responden sebanyak 800 orang pelaku bisnis yang terdiri dari *chief marketing officers, chief strategy officers, senior marketer dan senior sales executives* diperkirakan pada tahun 2020 hampir 80% bisnis akan menggunakan *chatbot* untuk meningkatkan pendapatan mereka [2]. Beberapa perusahaan yang sekarang menggunakan *chatbot* seperti, BCA dengan Vira, Unilever dengan Jemma atau Telkomsel dengan Veronica. Saat sebuah perusahaan mengaplikasikan *chatbot* untuk menawarkan layanan konsumen yang lebih baik, perusahaan dapat menghemat biaya operasional, selain itu dengan adanya *chatbot* perusahaan dapat menyediakan layanan yang aktif selama 24 jam. *Chatbot* dapat membantu suatu bisnis untuk berinteraksi dengan banyak konsumen dalam waktu yang

bersamaan tanpa perlu campur tangan dari *customer service* yang harus menanggapi pertanyaan - pertanyaan konsumen secara manual.

Oleh karena itu, tugas akhir ini menawarkan *chatbot* yang akan diimplementasikan pada Gifood.id. Gifood adalah platform yang menghubungkan orang-orang yang memiliki makanan berlebih dengan mereka yang membutuhkannya [3]. *Chatbot* berbasis web, dimana pengguna dapat mencari makanan, berbagi makanan, dan mencari *volunteer* untuk membagikan makanan. Salah satu contoh implementasi *chatbot* pada Gifood.id adalah pengguna mungkin menginginkan informasi tambahan mengenai makanan yang mereka inginkan mereka bisa langsung berinteraksi dengan *chatbot* tersebut. Di dalam *chatbot* ini juga dilengkapi dengan *speech recognition* dimana pengguna dapat berbicara dengan *chatbot*. Diharapkan dengan adanya *chatbot* ini pengguna dapat lebih efisien, mudah, dan praktis dalam menggunakan layanan dari Gifood.id, selain itu pengguna dapat berinteraksi dengan *chatbot* layaknya berbicara dengan manusia memakai bahasa sehari – hari.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membangun *chatbot* pada aplikasi web untuk Gifood.id ?
2. Bagaimana cara implementasi *speech recognition* untuk mendapatkan ucapan pengguna pada *chatbot*?
3. Bagaimana cara implementasi *Natural Language Processing* (NLP) pada *chatbot* ?
4. Bagaimana cara mengevaluasi *chatbot* yang dibangun ?

1.3 Batasan Masalah

Batasan masalah pada Tugas Akhir ini antara lain:

1. *Chatbot* dapat dijalankan pada *platform* web.
2. *Chatbot* ini berfungsi layaknya seperti *customer service* yang menanggapi *request* dari pengguna.

3. Bahasa dasar *chatbot* adalah Bahasa Indonesia.
4. Pengguna dapat berinteraksi dengan *chatbot* menggunakan *speech recognition* untuk mengenali ucapan pengguna dan *chatbot* akan menanggapi ucapan dari pengguna.
5. Jawaban dari *chatbot* berupa tulisan.
6. Pertanyaan harus minimal terdiri dari dua jenis kata dan tidak boleh terdiri dari 1 frasa.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain:

1. Membangun *chatbot* pada *platform* web sebagai pelayanan otomatis dan optimal kepada pengguna untuk membangun suasana layaknya berbicara dengan manusia dan mengembalikan jawaban yang tepat.
2. Mengimplementasikan *speech recognition* untuk mengenali ucapan pengguna.
3. Mengimplementasikan *Natural Language Processing* (NLP) untuk memproses pertanyaan pengguna sehingga *chatbot* akan menanggapi sesuai dengan ucapan pengguna.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah memberikan sebuah layanan kepada pengguna Gifood.id dalam mencari informasi tentang Gifood, berbagi makanan ataupun mencari *volunteer* untuk membagikan makanan. Dengan adanya Tugas Akhir ini, pengguna dapat lebih efisien, mudah, dan praktis dalam menggunakan layanan Gifood.id, selain itu pengguna dapat berinteraksi dengan *chatbot* layaknya berbicara dengan manusia memakai bahasa sehari – hari.

1.6 Metodologi Pembuatan Tugas Akhir

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

1. Penyusunan proposal Tugas Akhir
Proposal Tugas Akhir ini berisi tentang penjelasan mengenai

pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri dari hal yang melatar belakangi tugas akhir, rumusan masalah yang diangkat, batasan masalah yang ada, tujuan dan manfaat dari tugas akhir ini. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung dalam pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusun proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. Studi literatur

Pada studi literatur ini akan dipelajari sejumlah referensi yang relevan terhadap tugas akhir yang akan dikerjakan, seperti mempelajari referensi paper yang diadaptasi oleh tugas akhir ini, mempelajari pembuatan *chatbot*, mempelajari mengenai teknologi *speech recognition*, mempelajari tentang implementasi *Natural Language Processing*, dan bagaimana mengintegrasikan ketiganya. Studi literatur ini didapatkan dari buku, jurnal, internet serta materi-materi kuliah yang berhubungan dengan sistem yang akan dibangun.

3. Analisis dan desain sistem

Pada tahap ini dilakukan analisis dan pendefinisian kebutuhan sistem untuk masalah yang dihadapi, terutama analisis terkait bagaimana *chatbot* bisa memahami bahasa sehari - hari manusia. Selanjutnya, dilakukan perancangan sistem dengan beberapa tahap sebagai berikut:

- a. Perancangan ontologi sebagai penyimpanan data.
- b. Perancangan chatbot, yakni implementasi *chatbot*, *speech recognition* dan *Natural Language Processing*.
- c. Perancangan lingkungan dan antarmuka aplikasi.
- d. Perancangan diagram kasus penggunaan sistem.

4. Implementasi

Pada tahap ini akan dilakukan pembangunan sistem. Sistem yang dimaksud disini, yaitu *chatbot* yang dibangun pada platform web dengan bahasa HTML, PHP, Javascript, Phyton dan dilengkapi dengan *speech recognition*.

5. Uji coba dan evaluasi

Pada tahap ini dilakukan pengujian sistem kepada pengguna secara langsung. Pengujian dan evaluasi sistem dilakukan untuk mengevaluasi hasil analisis program. Tahapan-tahapan dari pengujian adalah sebagai berikut:

- a. Pengujian terhadap fungsionalitas yang terdapat di aplikasi *chatbot*.
- b. Hasil pengujian berupa evaluasi sistem, diukur melalui kuesioner tentang kepuasan *chatbot*.

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.

Bab VI 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 yang berisi kode sumber dan pertanyaan kuisisioner.

[Halaman ini sengaja dikosongkan]

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 Literature Review Penelitian Terdahulu

Tabel 2.1 Literature Review Penelitian Terdahulu

Judul Paper atau Penelitian (tahun)	Deskripsi garis besar	Kelebihan	Kekurangan
Implementasi Perintah Menampilkan Data Menggunakan Bahasa Indonesia Dengan <i>Natural Language Processing</i> (2013)	Merubah bahasa alami menjadi query sql	Dapat merubah bahasa alami menjadi format query sql dan menampilkan hasil query	Format input harus sesuai dengan variabel pada query
<i>Home Automation using IoT and a Chatbot using Natural Language Processing</i> (2017)	Oomatisasi rumah - mengendalikan kipas, lampu, dan peralatan listrik lain di rumah menggunakan Internet	Dapat menkontrol atau menyalakan dan mematikan semua peralatan elektronik yang ada dirumah.	Sebelum menggunakan <i>chatbot</i> , harus memasukkan jumlah ruangan dan peralatan rumah tangga. Tidak banyak variasi input.

Judul Paper atau Penelitian (tahun)	Deskripsi garis besar	Kelebihan	Kekurangan
<i>OntBot : Ontology based ChatBot (2011)</i>	OntBot menggunakan pemetaan untuk mengubah ontologi dan pengetahuan menjadi relasional database.	Mengubah bahasa alami yang sudah ditentukan format kelimatnya menjadi query.	Pola kalimat kurang variasi dan pengguna hanya dapat mengubah untuk objeknya.
<i>Ontology-Driven Personalized Food and Nutrition Planning System for the Elderly (2012)</i>	Menjelaskan pengembangan dan desain nutrisi dan perencanaan makanan yang dipersonalisasi sistem untuk orang tua.	Memetakan makanan yang nantinya akan digunakan untuk mendapatkan makanan yang sesuai dengan kondisi orang tua.	Kurangnya otomasi untuk pembelian dan inventaris persediaan makanan.
<i>Chatbot for E-Learning: A Case of Study (2018)</i>	Chatbot yang digunakan di bidang Pendidikan yaitu e-learning untuk membantu mahasiswa dalam urusan perkuliahan.	Memberikan saran atau jawaban dalam bidang dasar – dasar komputer dan jaringan komputer.	Akan dikembangkan melibatkan penerapan yang diusulkan pendekatan dalam berbagai konteks dan peningkatan platform elearning

Dari *literature* yang sudah ada, pada tugas akhir ini akan dibuat sebuah *chatbot* sebagai layanan dari Gifood.id yang dapat

menjawab pertanyaan - pertanyaan seputar Gifood.id yang sebelumnya belum ada pada layanan Gifood.id. *Chatbot* akan dibangun menggunakan *Natural Language Processing*.

Melihat *literature* penelitian sebelumnya, kelebihan *chatbot* yang akan dibangun mampu merubah pertanyaan dari bahasa alami menjadi query sparql dengan beberapa variasi pola kalimat, chatbot juga dilengkapi dengan speech recognition yang dapat mempermudah pengguna dalam bertanya dan membuat suasana yang lebih interaktif seperti berbicara dengan manusia. Selain menjawab pertanyaan dari pengguna, chatbot ini juga mampu menyimpan data ke dalam file RDF.

2.2 Chatbot

Untuk memberikan jawaban yang sesuai untuk kata kunci atau frasa yang diambil dari ucapan dan untuk menjaga percakapan terus menerus, ada kebutuhan untuk membangun sistem dialog (program) yang disebut *Chatbot* (Obrolan-Bot). *Chatbot* dapat membantu dalam interaksi komputer manusia dan mereka memiliki kemampuan untuk memeriksa dan mempengaruhi perilaku pengguna dengan mengajukan pertanyaan dan menanggapi pertanyaan-pertanyaan pengguna. *Chatbot* adalah program komputer yang meniru percakapan cerdas. Masukan ke program ini adalah teks bahasa alami, dan aplikasi harus memberikan jawaban yang merupakan jawaban cerdas terbaik untuk kalimat masukan. Proses ini diulang saat percakapan berlanjut dan responsnya berupa teks atau ucapan. Membangun *Chatbot* membutuhkan keterampilan pemrograman yang sangat profesional dan pengembang berpengalaman untuk mencapai bahkan tingkat realisme dasar. Ada platform pengembangan yang rumit di belakang *Chatbot* yang hanya akan sebagus basis pengetahuannya yang memetakan kata-kata pengguna ke dalam respons yang paling tepat. Pengembang bot biasanya membangun basis pengetahuan juga. Namun, ada beberapa platform yang menyediakan lingkungan belajar. Menulis *Chatbot* yang sempurna sangat sulit karena membutuhkan basis data yang sangat besar dan

harus memberikan jawaban yang masuk akal untuk semua interaksi. Ada sejumlah pendekatan untuk menciptakan basis pengetahuan untuk *Chatbot* dan termasuk menulis dengan tangan dan belajar dari korpus. Belajar di sini berarti menyimpan frasa baru dan kemudian menggunakannya nanti untuk memberikan jawaban yang tepat untuk frasa serupa. Merancang paket perangkat lunak *Chatbot* membutuhkan identifikasi bagian-bagian konstituen. Sebuah *Chatbot* dapat dibagi menjadi tiga bagian: *Responder*, *Classifier* dan *Graphmaster*, yang akan dijelaskan sebagai berikut:

- 1) *Responder*: itu adalah bagian yang memainkan peran interfacing antara rutinitas utama bot dan pengguna. Tugas responden adalah: mentransfer data dari pengguna ke Pengklasifikasi dan mengontrol *input* dan *output*.
- 2) *Classifier*: itu adalah bagian antara *Responder* dan *Graphmaster*. Fungsi lapisan ini adalah: memfilter dan menormalkan *input*, menyegmentasikan *input* yang dimasukkan oleh pengguna ke dalam komponen logis, mentransfer kalimat yang dinormalisasi ke dalam *Graphmaster*, memproses output dari *Graphmaster*, dan menangani instruksi dari sintaks basis data (misal AIML).
- 3) *Graphmaster*: adalah bagian untuk pencocokan pola yang melakukan tugas-tugas berikut: mengatur isi otak, menyimpan, dan menahan algoritma pencocokan pola [4] [5].

2.3 Natural Language Processing

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari user bisa dipahami dengan baik oleh komputer. [6]

Natural Language Processing dari sebuah teks terdiri dari 3 fase, antara lain :

1. Analisis morfologis

Untuk setiap hubungan alolog untuk kategori kata seperti jenis kelamin, kasus, kemunduran dan sebagainya secara resmi diambil dari sebuah kalimat.

2. Analisis sintaksis

Hubungan sintaksis antara kata-kata dibuat dalam kalimat, utama dan bagian kalimat bawahan diekstraksi, jenis kalimatnya adalah ditentukan, dan sebagainya. Analisis sintaksis dieksekusi tahap demi tahap menggunakan data diperoleh pada langkah sebelumnya. Fase ini menggunakan leksikal dan aturan sintaksis untuk menganalisis bahasa.

3. Analisis semantik

Fase analisis semantik adalah fase yang paling sulit pada *natural language processing*. Analisis semantik didasarkan pada mesin pengetahuan untuk bidang subjek tertentu dan informasi yang diterima dalam fase sebelumnya. Pada tahap ini konstruksi linguistik dibandingkan dengan konstruksi yang disimpan dalam memori sistem. Membuat model kata semantik disebut sebagai polisemantisisme. *Word sense* adalah serangkaian nilai yang mungkin, masing-masing dari mereka dapat diimplementasikan dalam bidang subjek tertentu. [7]

Natural Language Processing digunakan untuk mengolah pertanyaan dari pengguna yang nantinya akan dirubah ke dalam format query sparql.

2.3.1 Tokensisasi

Kalimat-kalimat dan frasa-frasa/klausa-klausa yang telah diperoleh dari tahap sebelumnya, dilakukan pemecahan menjadi token-token (*tokenization*) atau kata-kata, dengan menggunakan karakter spasi atau tanda baca sebagai pemisah. Pemecahan

kalimat ini menggunakan fungsi dari *Natural Language Toolkit* (NLTK). [8]

Tokenisasi digunakan untuk memecah kalimat pertanyaan dari pengguna ke dalam token – token yang nantinya akan digunakan untuk menentukan jenis kata dan kata dasar.

2.3.2 Lemma

Lemmatisasi adalah proses pengelompokan bersama berbagai bentuk kata yang berbeda sehingga dapat dianalisis sebagai satu item. Lemmatisasi mirip dengan *stemming* tetapi membawa konteks kata-kata. Jadi itu menghubungkan kata-kata dengan makna yang mirip dengan satu kata. *Lemma* dapat mengacu kepada beberapa hal berikut: lema (linguistik) - bentuk baku dari sebuah kata, entri kata dalam kamus. [9]

Lemma digunakan untuk mendapatkan kata dasar dari kalimat pertanyaan pengguna. Kata dasar ini nantinya akan digunakan untuk menyusun pola *regular expression*.

2.3.3 POS Tag

POS (Part-of-Speech) Tag merupakan suatu cara untuk pengkategorian kelas kata, seperti kata benda, kata kerja, kata sifat, dll. *POS Tagger* merupakan sebuah aplikasi yang mampu melakukan proses anotasi *part-of-speech tag* untuk setiap kata dalam sebuah dokumen secara otomatis. [10]

POS Tag digunakan untuk mendapatkan jenis kata dari kalimat pertanyaan pengguna. Kata dasar ini nantinya akan digunakan untuk menyusun pola *regular expression*.

2.4 Speech Recognition

Speech Recognition (juga dikenal sebagai *Automatic Speech Recognition* (ASR), atau pengenalan suara komputer) adalah proses mengubah sinyal suara ke urutan kata, dengan menggunakan algoritma yang diimplementasikan sebagai program komputer. Penelitian dalam pengolahan suara dan komunikasi untuk sebagian besar, dimotivasi oleh keinginan orang untuk

membangun model mekanik untuk meniru kemampuan komunikasi verbal manusia.

Pidato adalah bentuk komunikasi yang paling alami dari seorang manusia dan pengolahan ucapan telah menjadi salah satu area yang paling menarik dari pemrosesan sinyal. Teknologi pengenalan suara telah memungkinkan komputer untuk mengikuti perintah suara manusia dan memahami bahasa manusia. Tujuan utama dari area pengenalan suara adalah untuk mengembangkan teknik dan sistem untuk input ucapan ke mesin. Teknologi pengenalan suara semakin banyak digunakan dalam jaringan telepon untuk mengotomatiskan serta untuk meningkatkan layanan operator. Laporan ini meninjau sorotan utama selama enam dekade terakhir dalam penelitian dan pengembangan pengenalan ucapan otomatis, sehingga memberikan perspektif teknologi. Meskipun banyak kemajuan teknologi telah dibuat, masih tetap ada banyak masalah penelitian yang perlu diatasi [11].

Speech recognition digunakan untuk mengenali ucapan dari pengguna. Ucapan pengguna yang berupa suara akan diubah kedalam teks atau tulisan yang nantinya akan digunakan sebagai *input* pada *Natural Language Processing*.

2.5 *Natural Language Toolkit*

Natural Language Toolkit (NLTK) adalah platform terkemuka untuk membangun program Python untuk bekerja dengan data bahasa manusia. Ini menyediakan antarmuka yang mudah digunakan untuk lebih dari 50 korpora dan sumber daya leksikal seperti WordNet, bersama dengan serangkaian pustaka pemrosesan teks untuk klasifikasi, *tokenization*, *stemming*, *tagging*, *parsing*, dan penalaran semantik, pembungkus untuk perpustakaan NLP yang berkekuatan industri, dan forum diskusi aktif.

NLTK telah disebut sebagai alat mengajar, dan bekerja dalam, linguistik komputasi menggunakan Python, dan perpustakaan yang luar biasa untuk bermain dengan bahasa alami. *Natural Language Processing* dengan Python menyediakan

pengantar praktis untuk pemrograman pemrosesan bahasa. Ditulis oleh pencipta NLTK, ia membimbing pembaca melalui dasar-dasar penulisan program Python, bekerja dengan korpora, mengkategorikan teks, menganalisis struktur linguistik, dan banyak lagi. [8]

Dalam pembangunan *Natural Language Processing*, banyak menggunakan fungsi dari *Natural Language Toolkit* (NLTK). Fungsi – fungsi yang digunakan antara lain: tokenisasi, pos tag, lemma, dll.

2.6 Quepy

Quepy adalah kerangka kerja python untuk mentransformasikan pertanyaan bahasa alami menjadi query dalam bahasa query database. Itu dapat dengan mudah disesuaikan untuk berbagai jenis pertanyaan dalam bahasa alami dan permintaan basis data. Jadi, dengan sedikit pengkodean dapat membangun sistem sendiri untuk akses bahasa alami ke database. Saat ini Quepy memberikan dukungan untuk bahasa query Sparql dan MQL [12].

Kerangka kerja quepy digunakan untuk membangun *Natural Language Processing*.

2.7 Ontologi

Ontologi adalah spesifikasi yang jelas tentang serangkaian konsep yang menjelaskan sebuah wilayah pengetahuan tertentu yang dipakai bersama-sama oleh para pengguna sistem (Pendit, 2008). Seringkali ontologi didefinisikan sebagai suatu struktur hirarkis yang mengandung definisi kelas (*class*), antar hubungan (*relationships*), karakteristik dan properti, dan tata aturan (*rules*) yang berlaku pada suatu bidang pengetahuan. Itulah sebabnya Gruber (1995) mendefinisikan ontologi sebagai sebuah spesifikasi yang formal dan eksplisit tentang konsep yang dipakai bersama (*a formal, explicit spesification of a shared conceptualization*).

Bahasa ontologi adalah sebuah bahasa formal dari suatu pengembangan ontologi. Beberapa bahasa pendukung teknologi semantic web yang digunakan dan menjadi struktur ontologi antara

lain XML, XMLS, RDF, RDFS, dan OWL. XML merupakan suatu perkembangan teknologi dan generasi baru penerus HTML. XML lebih berkonsentrasi pada struktur data. Dalam hal ini XML sebagai pioner yang digunakan untuk membawa data dan dapat digunakan untuk mengirim data sehingga pertukaran data akan lebih mudah [13].

Ontologi digunakan untuk menyimpan data makanan, kota, transaksi makanan yang dibagikan, dan informasi tentang Gifood. Data ini digunakan untuk menjawab pertanyaan – pertanyaan pengguna.

2.8 RDF

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

RDF menggunakan URI untuk mengidentifikasika 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 sumber daya web, sehingga dapat dibaca dan dipahami oleh aplikasi komputer.

RDF dapat digunakan dalam beberapa aplikasi, salah satu yang paling penting adalah dapat mencari sumber daya pengetahuan. RDF dapat digunakan untuk meningkatkan kemampuan dari mesin pencari. RDF juga digunakan untuk memfasilitasi berbagai pengetahuan dan perukaran perangkat lunak cerdas, dan seperti yang disebutkan sebelumnya, untuk

menggambarkan konten dan hubungan konten yang tersedia dengan sumber daya apapun, seperti halaman (*page*).

RDF memiliki 3 elemen, subjek, objek, dan predikat. Dimungkinkan bahwa <subjek> memiliki properti <predikat> dilanjutkan oleh <objek>. [13]

RDF digunakan untuk mendefinisikan kelas, *data property*, *object property* dari ontologi yang akan digunakan.

2.9 RDFlib

RDFlib adalah *library* Python yang khusus untuk handle RDF. RDFlib berisi sebagian besar hal yang dibutuhkan untuk bekerja dengan RDF, diantaranya adalah [14]:

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 digunakan untuk membantu proses query sparql untuk *file rdf*.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan ini meliputi analisis metode secara umum dan perancangan sistem.

3.1 Analisis Metode Secara Umum

Pada subbab berikut akan dijelaskan tentang penggunaan *Natural Language Processing* pada *chatbot*.. Analisis yang dilakukan meliputi analisis permasalahan, deskripsi umum sistem, dan kebutuhan fungsional sistem.

3.1.1 Analisis Permasalahan

Permasalahan utama yang diangkat pada pembuatan Tugas Akhir ini adalah bagaimana implementasi *Natural Language Processing* pada *chatbot* untuk mengolah dari bahasa alami menjadi sebuah query untuk mendapatkan jawaban sesuai dengan pertanyaan pengguna.

Chatbot merupakan salah satu hal populer yang digunakan diberbagai perusahaan. *Chatbot* sendiri banyak digunakan oleh banyak perusahaan. Umumnya *chatbot* dipadukan dengan *Natural Language Processing* (NLP) dan *Machine Learning* (ML) untuk mampu memahami dan merespon *request* dari para pengguna. Gifood merupakan salah satu *start-up* yang sedang berkembang ingin mengembangkan *chatbot* sebagai salah satu layanan bagi penggunanya.

Untuk mengatasi permasalahan tersebut, pada Tugas Akhir ini akan dilakukan pembuatan *chatbot* menggunakan *Natural Language Processing* untuk melayani pengguna yang ingin tahu tentang Gifood, mencari makanan yang tersedia, berbagi makanan berlebih maupun mencari *volunteer* untuk membagikan makanan yang berlebih. *Chatbot* ini juga dilengkapi dengan *speech recognition* untuk menambah interaksi dengan pengguna seolah

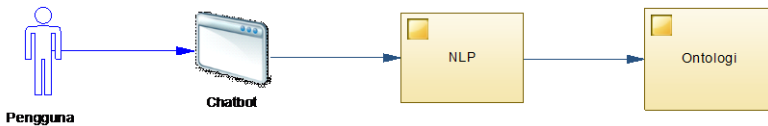
olah sedang berbicara dengan manusia. *Chatbot* ini dikembangkan dalam *platform* web.

Pada Tugas Akhir ini lebih difokuskan tentang implementasi *Natural Language Processing* yang digunakan untuk mengubah bahasa alami menjadi query untuk menjawab pertanyaan pengguna.

3.1.2 Deskripsi Umum Sistem

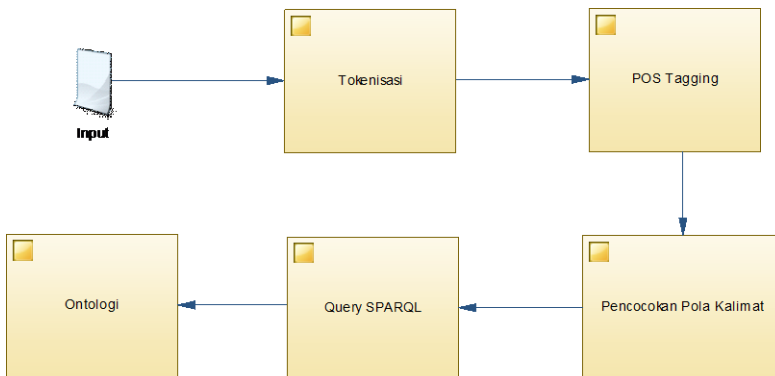
Pada Tugas Akhir ini akan dibuat sebuah *chatbot* yang digunakan untuk menjawab pertanyaan pengguna seputar layanan Gifood. *Chatbot* ini dilengkapi dengan *speech recognition* untuk menambah interaksi dengan pengguna. Untuk menggunakan *chatbot* ini, perangkat harus terhubung dengan internet.

Pada Gambar 3.1 dijelaskan bahwa pertanyaan pengguna yang berupa bahasa alami akan di proses menggunakan *Natural Language Processing* menjadi query untuk mendapatkan jawaban dari data yang dimiliki oleh Gifood.



Gambar 3.1 Deskripsi Umum Sistem

Sedangkan pada *Natural Language Processing* data akan diproses mulai dari tokenisasi, *pos tagging*, *parsing*. Nilai dari *parsing* yang sesuai dengan pola atau kata kunci yang telah ditentukan akan dirubah menjadi query untuk mendapatkan jawaban dari pertanyaan tersebut. Tahapan pada *Natural Language Processing* digambarkan pada Gambar 3.2.



Gambar 3.2 Tahapan pada *Natural Language Processing*

3.1.2.1 *Speech Recognition*

Pengguna dapat bertanya ke *chatbot* dengan menuliskan pertanyaan pada kolom pertanyaan, selain itu pengguna juga dapat menggunakan *speech recognition*. Sistem ini menggunakan *HTML5 Speech Recognition API*.

3.1.2.2 **Tokenisasi**

Sebelum melakukan proses perubahan menjadi query untuk menjawab pertanyaan dari pengguna, terlebih dahulu dilakukan proses tokenisasi. Hal ini dilakukan karena dalam penentuan pola kalimat dibutuhkan jenis per kata dalam satu kalimat. Fungsi yang digunakan untuk proses tokenisasi adalah fungsi dari *Natural Language Processing Toolkit (NLTK)*. Contoh proses tokenisasi dijelaskan pada Tabel 3.1.

Tabel 3.1 Contoh Hasil Tokenisasi

Pertanyaan	Apa itu Gifood ?
Hasil Tokenisasi	['Apa', 'itu', 'Gifood', '?']

3.1.2.1 POS Tagger

Pertanyaan dari pengguna yang telah diubah menjadi token – token dari proses tokenisasi akan ditentukan jenis kata tersebut dengan fungsi *Part-of-Speech Tag* (POS Tag). Fungsi yang digunakan berasal dari *Natural Language Processing Toolkit* (NLTK). Fungsi *POS Tag* pada NLTK masih menggunakan Bahasa Inggris, jadi untuk melakukan proses ini pertanyaan dari pengguna akan diubah terlebih dahulu ke dalam Bahasa Inggris menggunakan API *Google Translate*. Contoh proses POS Tag dijelaskan pada Tabel 3.2. Dan jenis – jenis dari POS Tag dapat dilihat pada Tabel 10.1.

Tabel 3.2 Contoh Hasil Pos Tag

Pertanyaan	Apa itu Gifood ?
Pertanyaan (Bahasa Inggris)	What is Gifood ?
Hasil POS Tag	[('What', 'WP'), ('is', 'VBZ'), ('Gifood', 'NNP'), ('?', '.')]]

3.1.2.2 Pencocokan Pola Kalimat

Pertanyaan yang telah ditentukan masing – masing jenis katanya, akan di cocokkan dengan fungsi yang berisi pola – pola kalimat yang telah ditentukan. Pola kalimat ini juga akan menentukan relasi apa saja yang akan digunakan untuk query pada tahap selanjutnya.

3.1.2.3 Pengubahan Kalimat ke dalam Bentuk Query Sparql

Dalam pengubahan bahasa alami ke dalam bentuk query, dilakukan dengan terlebih dahulu menggunakan bentuk khusus dari ekspresi reguler. Kemudian ditentukan dengan relasi – relasi yang akan digunakan. Jenis – jenis relasi yang akan digunakan untuk menangani seluruh kebutuhan fungsional telah ditentukan sebelumnya. Contoh pengubahan kalimat ke dalam bentuk query dijelaskan pada Tabel 3.3.

Tabel 3.3 Contoh Proses Perubahan dari Bahasa Alami Menjadi Query Sparql

Pertanyaan	Apa itu Gifood ?
Ekspresi regular	Lemma("what") + Lemma("be") + Question(Pos("DT")) + Thing() + Question(Pos("."))
Relasi	HasKeyword(), DefinitionOf()
Query	SELECT DISTINCT ?x1 WHERE { ?x0 rdfs:label "Gifood"@en. ?x0 ta:About ?x1. }

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Subbab ini membahas spesifikasi kebutuhan perangkat lunak dari hasil analisis yang telah dilakukan. Bagian ini berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk kebutuhan fungsional, kebutuhan non fungsional, dan diagram kasus penggunaan.

3.1.3.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan yang harus dimiliki oleh perangkat lunak, pertanyaan yang dapat ditangani oleh *chatbot*, reaksi dari perangkat lunak terhadap suatu masukan, hasil yang dilakukan perangkat lunak pada situasi khusus. Kebutuhan fungsional dari perangkat lunak dijelaskan pada Tabel 3.4.

Tabel 3.4 Kebutuhan Fungsional

Kode	Kebutuhan Fungsional	Deskripsi
F01	Jenis makanan yang dapat di bagikan.	Pengguna dapat bertanya jenis makanan apa saja yang dapat dibagikan.
F02	Apa itu Gifood.	Pengguna dapat bertanya tentang apa itu Gifood.
F03	Cara kerja Gifood	Pengguna dapat bertanya tentang cara kerja Gifood.

Kode	Kebutuhan Fungsional	Deskripsi
F04	<i>Contact Person Gifood</i>	Pengguna dapat bertanya tentang <i>contact person</i> Gifood yang dapat dihubungi.
F05	Mitra Gifood	Pengguna dapat bertanya tentang mitra sosial dan mitra usaha Gifood.
F06	Cara mengetahui kelayakan makanan	Pengguna dapat bertanya tentang bagaimana cara mengetahui kelayakan makanan yang ingin dibagikan.
F07	Apa itu <i>Food Warrior</i>	Pengguna dapat bertanya tentang apa itu <i>Food Warrior</i> .
F08	Cara menjadi <i>Food Warrior</i>	Pengguna dapat bertanya tentang bagaimana cara menjadi <i>Food Warrior</i> .
F09	Mencari <i>Volunteer</i>	Pengguna dapat mencari <i>volunteer</i> untuk membagikan makanan.
F10	Makanan berlebih	Pengguna dapat bertanya tentang bagaimana jika makanan yang sudah dibagikan masih tersisa atau berlebih.
F11	Mencari makanan yang tersedia dengan nama, lokasi, waktu, status ketersediaan, dan jumlah porsi.	Pengguna dapat mencari semua makanan yang tersedia atau mencari menggunakan nama makanan, lokasi, waktu, status ketersediaan, dan jumlah porsi.
F12	Berbagi makanan berlebih	Pengguna dapat membagikan makanan.

3.1.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional mendefinisikan batasan layanan yang ditawarkan oleh sistem. Kebutuhan non fungsional dari perangkat lunak dijelaskan pada Tabel 3.5.

Tabel 3.5 Kualitas Perangkat Lunak

No	Parameter	Deskripsi
1	Bahasa	Bahasa yang digunakan pada antarmuka merupakan bahasa Indonesia
2	<i>Portability</i>	Aplikasi mudah untuk dioperasikan pada <i>browser</i> computer.
3	<i>Correctness</i>	Aplikasi menghasilkan keluaran yang benar untuk setiap kemungkinan masukan oleh <i>user</i>

3.1.3.3 Aktor

Pengertian pengguna adalah pihak-pihak, baik manusia maupun sistem atau perangkat lain yang terlibat dan berinteraksi secara langsung dengan sistem. Pada perangkat lunak ini hanya terdapat satu pengguna yaitu pelanggan Gifood.

3.1.3.4 Kasus Penggunaan

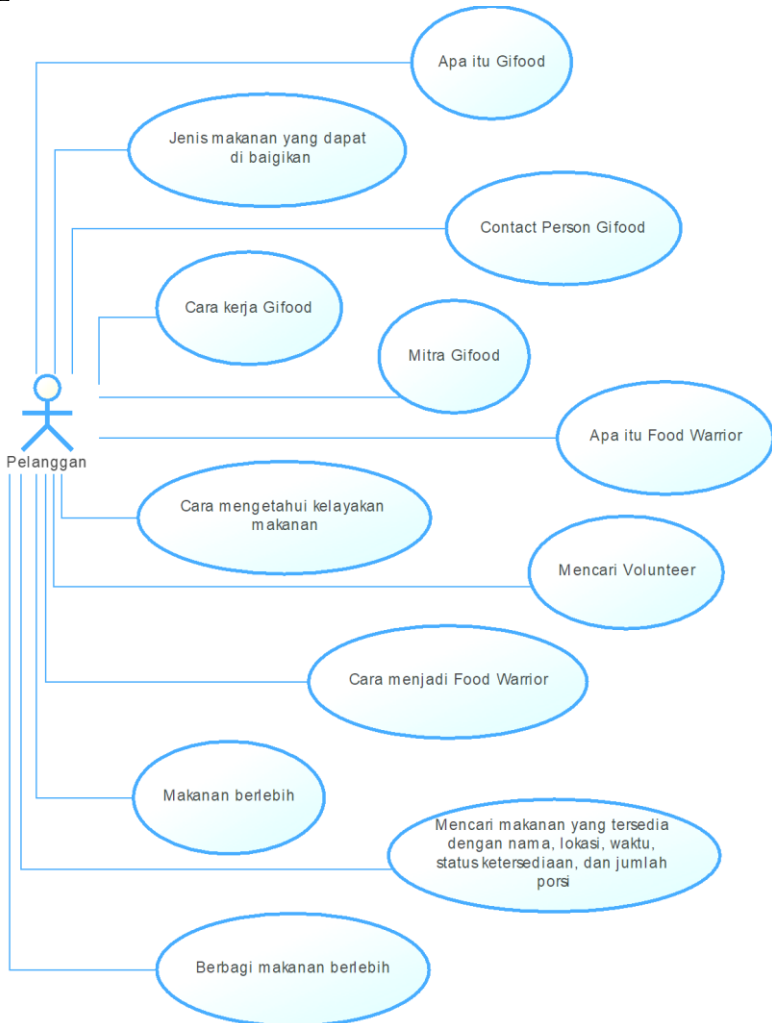
Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan dijelaskan pada Tabel 3.6 dan diagram kasus penggunaan ditunjukkan pada Gambar 3.3.

Tabel 3.6 Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Aktor
UC-001	Jenis makanan yang dapat di baigikan.	Pelanggan
UC-002	Apa itu Gifood.	Pelanggan
UC-003	Cara kerja Gifood	Pelanggan
UC-004	<i>Contact Person</i> Gifood	Pelanggan
UC-005	Mitra Gifood	Pelanggan

Kode Kasus Penggunaan	Nama	Aktor
UC-006	Cara mengetahui kelayakan makanan	Pelanggan
UC-007	Apa itu <i>Food Warrior</i>	Pelanggan
UC-008	Cara menjadi <i>Food Warrior</i>	Pelanggan
UC-009	Mencari <i>Volunteer</i>	Pelanggan
UC-010	Makanan berlebih	Pelanggan
UC-011	Mencari makanan yang tersedia dengan nama, lokasi, waktu, status ketersediaan, dan jumlah porsi	Pelanggan
UC-012	Berbagi makanan berlebih	Pelanggan

Z



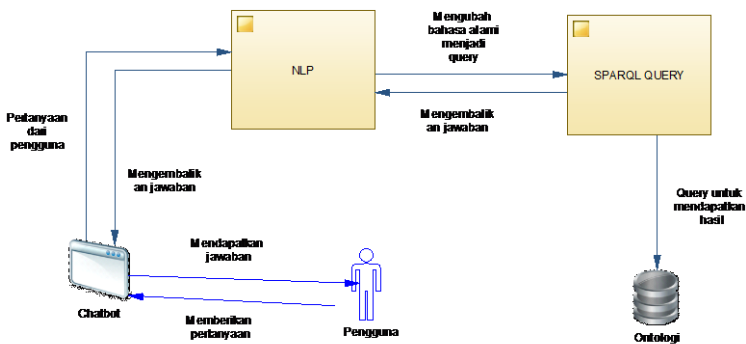
Gambar 3.3 Diagram Kasus Penggunaan

3.2 Perancangan

Dalam subbab ini membahas perancangan dari aplikasi Tugas Akhir. Subbab ini terdiri dari lingkungan perancangan perangkat lunak, perancangan arsitektur sistem, perancangan ontologi, dan perancangan antarmuka pengguna.

3.2.1 Perancangan Arsitektur Sistem

Arsitektur dari *chatbot* ini ditunjukkan pada Gambar 3.4.



Gambar 3.4 Arsitektur Sistem *Chatbot*

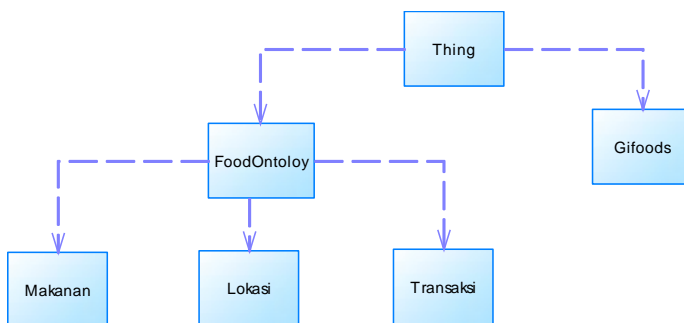
Pengguna dapat berinteraksi dengan chatbot melalui tampilan antarmuka *chatbot* yang dibangun pada platform website. Pengguna dapat bertanya kepada *chatbot*, dan *chatbot* akan membalas pertanyaan sesuai dengan pertanyaan dari pengguna. Pertanyaan tersebut akan di proses menggunakan *Natural Language Processing* (NLP) untuk mengubah bahasa alami (pertanyaan dari pengguna) ke dalam bentuk query sparql yang digunakan untuk mendapatkan jawaban dari data yang disimpan dalam bentuk file RDF.

3.2.2 Perancangan Ontologi

Dalam pembuatan Tugas Akhir ini, diperlukan analisis perancangan Ontologi. Alasan menggunakan ontologi dalam pembangunan sistem ini adalah:

1. Menjelaskan suatu domain pengetahuan secara eksplisit; memberikan struktur hierarki dari konsep untuk menjelaskan sebuah domain dan bagaimana mereka berhubungan.
2. 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.
3. 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. [15]

Data pada sistem ini akan disimpan ke dalam file RDF. Rancangan kelas ontologi ditampilkan pada Gambar 3.5 .



Gambar 3.5 *Conceptual Data Model Pada Database Plugin Meaningful Learning*

3.2.2.1 Daftar Kelas

Daftar kelas yang terdapat dalam ontologi ini antara lain : Food Ontology yang terdiri dari kelas Lokasi, Makanan, dan Transaksi, dan kelas Gifoods.

3.2.2.2 Daftar *Data Properties*

Data Properties yang terdapat dalam ontologi ini antara lain : About, Alamat, ContactPerson, Deskripsi, Expired_at, HowItWorks, HowToBe, JaminanMakananHabis, Judul, Jumlah, KelayakanMakanan, MakananBerlebih, Mitra, Status.

3.2.2.3 Daftar *Object Properties*

Object Properties atau relasi yang terdapat dalam ontologi ini antara lain : MemilikiMakanan, MempunyaiMitra, TerdapatDi.

3.2.3 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan dengan aktor harus memberikan kemudahan serta tampilan yang menarik bagi penggunanya. Pada *chatbot* ini, pengguna yaitu pelanggan dapat bertanya kepada *chatbot* dan *chatbot* akan menjawab pertanyaan tersebut, pengguna juga dapat membagikan makanan dengan mengisi form untuk membagikan makanan.

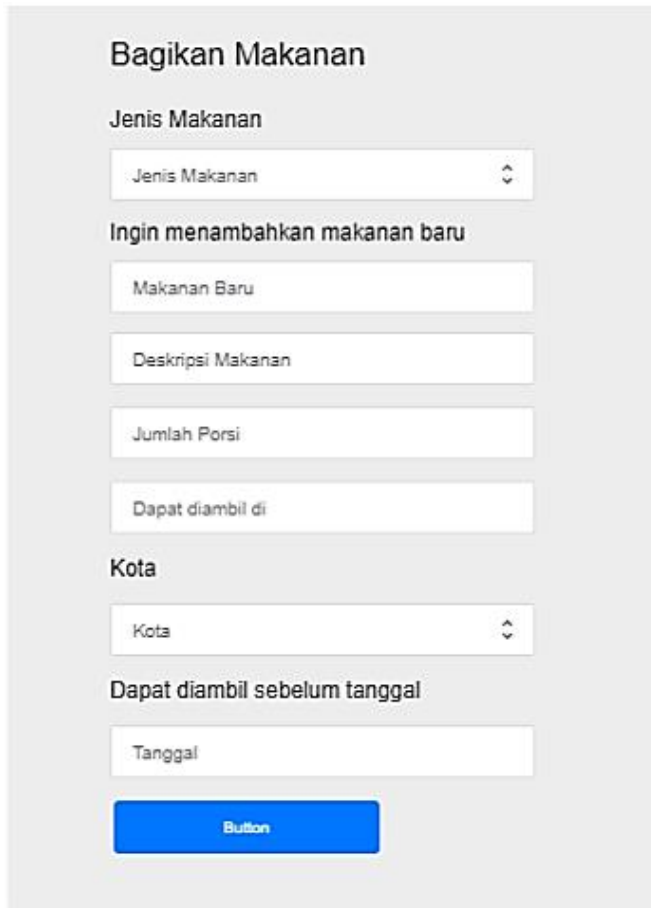
3.2.3.1 Rancangan Halaman Antarmuka *Chatbot*

Halaman ini digunakan oleh pengguna untuk bertanya kepada chatbot Gifood. Pada halaman ini terdapat kolom pertanyaan, tombol untuk berbagi makanan, dan tombol *speech recognition*. Rancangan halaman ditunjukkan pada Gambar 3.6.



Gambar 3.6 Rancangan Halaman Antarmuka Chatbot

3.2.3.2 Rancangan Halaman Antarmuka Form Membagikan Makanan



The image shows a web form titled "Bagikan Makanan" (Share Food). The form is set against a light gray background and contains the following elements:

- Jenis Makanan**: A dropdown menu with the placeholder text "Jenis Makanan" and a double-headed arrow icon.
- Ingin menambahkan makanan baru**: A section header followed by four input fields:
 - Makanan Baru**: A text input field.
 - Deskripsi Makanan**: A text input field.
 - Jumlah Porsi**: A text input field.
 - Dapat diambil di**: A text input field.
- Kota**: A dropdown menu with the placeholder text "Kota" and a double-headed arrow icon.
- Dapat diambil sebelum tanggal**: A section header followed by a text input field with the placeholder text "Tanggal".
- Button**: A blue rectangular button with the text "Button" centered on it.

Gambar 3.7 Rancangan Halaman Antarmuka Form Membagikan Makanan

Halaman ini digunakan oleh pengguna untuk membagikan makanan yang berlebih. Rancangan halaman ditunjukkan pada Gambar 3.7 dan atribut antarmuka ditunjukkan pada Tabel 3.7.

Tabel 3.7 Atribut Antarmuka Form Membagikan Makanan

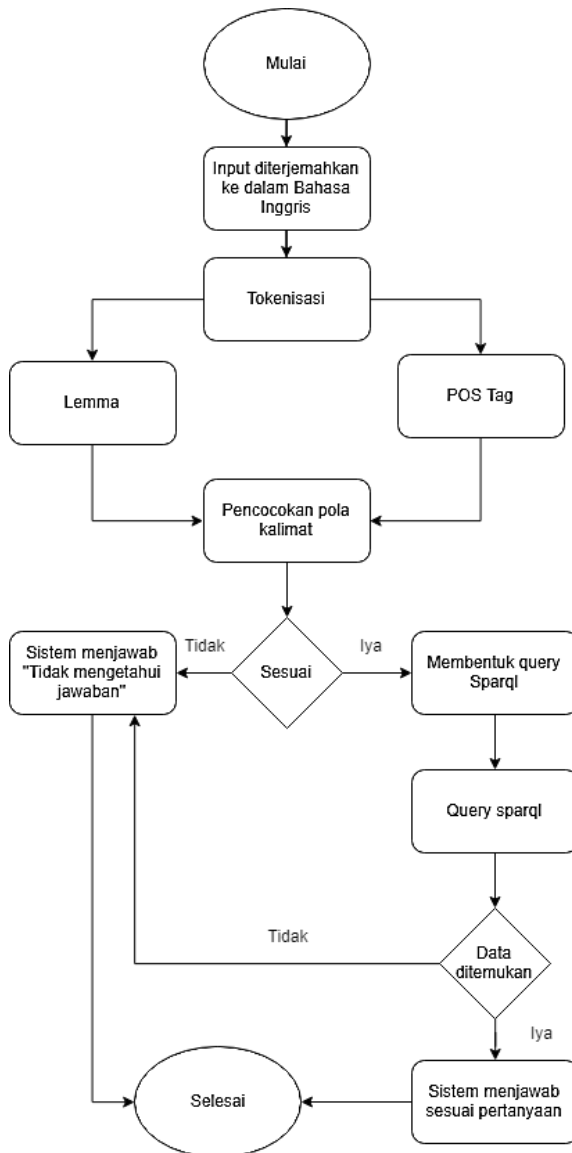
No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>jenisMaknan</i>	<i>Input text</i>	Berisi tentang nama makanan yang pernah dibagikan	<i>String</i>
2	<i>makananBaru</i>	<i>Input text</i>	Masukan nama makanan baru yang tidak terdapat pada jenis makanan	<i>String</i>
3	<i>deskripsiMakanan</i>	<i>Input text</i>	Deskripsi makanan yang ingin dibagikan	<i>String</i>
4	<i>jumlahPorsi</i>	<i>Input text</i>	Jumlah porsi	<i>String</i>
5	<i>alamat</i>	<i>Input text</i>	Alamat pengambilan	<i>String</i>
6	<i>kota</i>	<i>Input text</i>	Kota pengambilan	<i>String</i>
7	<i>tanggal</i>	<i>Input text</i>	Batas maksimal pengambilan makanan	<i>Date</i>

3.2.4 Diagram Alur Sistem

Pada Gambar 3.8, terdapat diagram alu sistem yang menjelaskan proses pengolahan mulai dari input sampai dengan output. Terdapat beberapa langkah yang dilakukan pada tahap ini, antara lain:

1. Menerjemahkan pertanyaan dari pengguna kedalam Bahasa Inggris.
2. Tokenisasi untuk memecah pertanyaan ke dalam token – token.

3. Lemmatisasi untuk menentukan masing – masing kata dasar dari pertanyaan pengguna.
4. POS Tag untuk mendapatkan masing – masing jenis kata dari pertanyaan pengguna.
5. *Parsing* hasil dari POS Tag dan Lemma kedalam fungsi Pos() dan Lemma() yang nantinya akan dicocokkan dengan pola yang telah dibuat dan menentukan predikat atau relasi yang akan digunakan.
6. Apabila pola tersebut cocok, akan dilakukan pembentukan query sparql sesuai dengan predikat atau relasi yang telah ditentukan sebelumnya. Apabila pola di cocok maka sistem akan menjawab “Tidak mengetahui jawaban”.
7. Melakukan proses query sparql menggunakan RDFlib.
8. Apabila data ditemukan maka sistem akan menjawab sesuai dengan pertanyaan pengguna. Apabila data tidak ditemukan maka sistem akan menjawab “Tidak mengetahui jawaban”.



Gambar 3.8 Diagram Alur Sistem

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

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

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

4.1 Lingkungan Implementasi Perangkat Lunak

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti ditampilkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

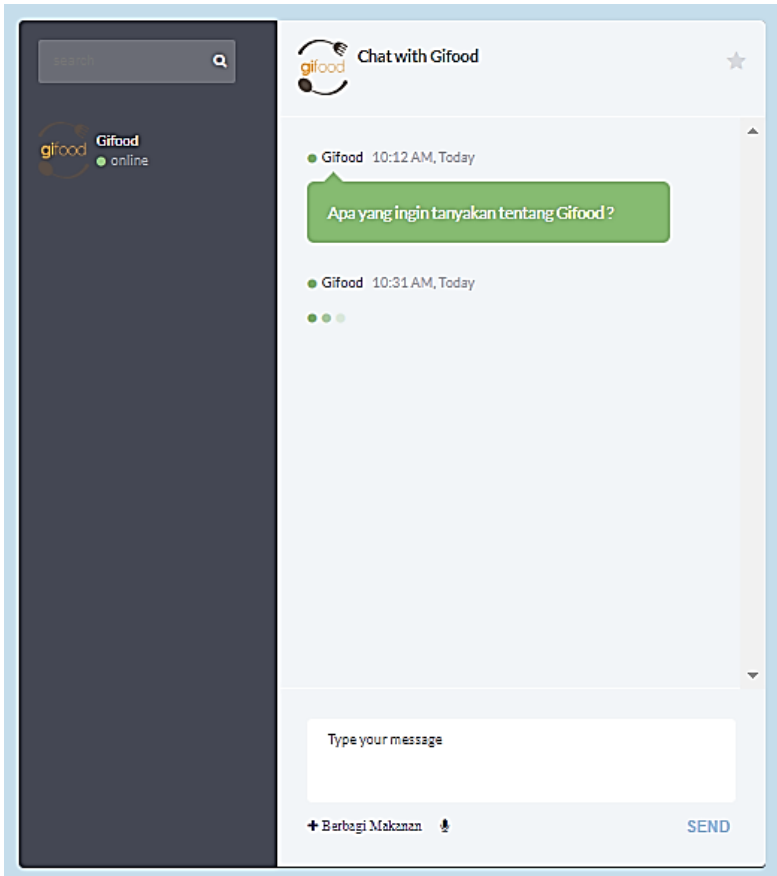
Perangkat Lunak	Sistem Operasi	Windows 10 Pro 64-bit
	Perangkat Lunak	Python 2.7.14, PHP 7.2.6, Protege-5.5.0-beta-3, Visual Studio Code, Microsoft Word 2016

4.2 Implementasi Antarmuka Pengguna

Subbab ini membahas tentang implementasi tampilan antarmuka yang telah dirancang dan dibahas pada Bab III. Selanjutnya akan dirinci berdasarkan halaman yang akan tampil dan dilihat oleh pengguna.

4.2.1 Implementasi Halaman *Chatbot*

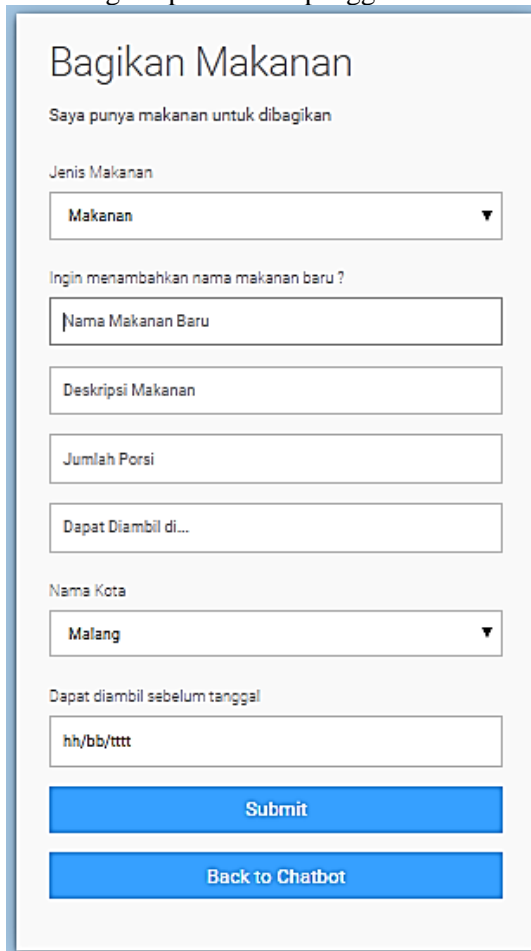
Halaman antarmuka *chatbot* pada Gambar 4.1 merupakan halaman yang digunakan oleh pengguna untuk bertanya seputar layanan Gifood. Halaman antarmuka ini mengacu pada kasus penggunaan nomor UC-001 sampai dengan UC-011. Terdapat *textarea* untuk memasukkan pertanyaan dari pengguna.



Gambar 4.1 Implementasi Halaman Chatbot

4.2.2 Implementasi Halaman Berbagi Makanan

Halaman antarmuka pengguna untuk membagikan makanan berupa *form* yang dapat dilihat pada Gambar 4.2. Halaman antarmuka ini mengacu pada kasus penggunaan nomor UC-012.



The image shows a web form titled "Bagikan Makanan" (Share Food). The form is designed for sharing food items and includes the following fields and controls:

- Title:** "Bagikan Makanan"
- Subtitle:** "Saya punya makanan untuk dibagikan" (I have food to share)
- Jenis Makanan:** A dropdown menu with "Makanan" selected.
- Ingin menambahkan nama makanan baru ?** A checkbox.
- Form Fields:** Four text input fields for "Nama Makanan Baru", "Deskripsi Makanan", "Jumlah Porsi", and "Dapat Diambil di...".
- Nama Kota:** A dropdown menu with "Malang" selected.
- Dapat diambil sebelum tanggal:** A date input field with the placeholder "hh/bb/tttt".
- Buttons:** A blue "Submit" button and a blue "Back to Chatbot" button.

Gambar 4.2 Implementasi Halaman Berbagi Makanan

4.3 Implementasi Fungsi Objek, Predikat, *Prefix*

4.3.1 Implementasi Fungsi Objek

Fungsi ini digunakan untuk inisiasi objek yang akan digunakan dalam query sparql pada tahap selanjutnya. Fungsi yang terdapat pada sistem ini antara lain : Thing(), Quantity(), Date(). Fungsi ini juga akan digunakan untuk mengembalikan nilai dari kata yang memiliki jenis kata seperti yang ada pada fungsi tersebut. Fungsi Thing() digunakan untuk jenis kata yang mengandung kata benda. Fungsi Quantity() digunakan untuk jenis kata yang mengandung jumlah. Fungsi Date() digunakan untuk jenis kata yang mengandung tanggal.

```

1. class Thing(Particle):
2.     regex = Plus(Pos("NN") | Pos("NNS") | Pos("NN
   P") | Pos("NNPS"))
3.
4.     def interpret(self, match):
5.         return HasKeyword(match.words.tokens)

```

Kode Sumber 4.1 Implementasi Fungsi Thing

```

1. class Quantity(Particle):
2.     regex = Plus(Pos("CD"))
3.
4.     def interpret(self, match):
5.         jumlah = match.words.tokens
6.         return jumlah

```

Kode Sumber 4.2 Implementasi Fungsi Quantity

```

1. class Date(Particle):
2.     regex = Plus( (Pos("CD")+Pos(":")+Pos("CD")+P
   os(":")+Pos("CD")) | (Pos("NNP")+ \
3.         Pos("CD")) | (Pos("NNP")+Pos(
   "CD")+Pos(",")+Pos("CD")) )
4.
5.     def interpret(self, match):
6.         tgl = match.words.tokens

```

```
7.         return tgl
```

Kode Sumber 4.3 Implementasi Fungsi Date

4.3.2 Implementasi Fungsi Predikat

Kode sumber dari fungsi predikat yang akan digunakan untuk membentuk query ditunjukkan pada Kode Sumber 8.1. Pada kode sumber ini terdapat fungsi yang digunakan untuk mengisi predikat pada query sparql.

4.3.3 Implementasi Fungsi Prefix

Prefix ini akan digunakan untuk proses query sparql dengan menggunakan fungsi dari *library rdflib*.

```
1. # Generated query language
2. LANGUAGE = "sparql"
3.
4. # NLTK config
5. NLTK_DATA_PATH = [] # List of paths with NLTK da
   ta
6.
7. # Encoding config
8. DEFAULT_ENCODING = "utf-8"
9.
10. # Sparql config
11. SPARQL_PREAMBLE = u"""
12. PREFIX owl: <http://www.w3.org/2002/07/owl#>
13. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-
   schema#>
14. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
   syntax-ns#>
15. PREFIX ta: <http://www.semanticweb.org/asus/ontol
   ogies/2019/1/untitled-ontology-49#>
16. """
```

Kode Sumber 4.4 Implementasi Fungsi Prefix

4.4 Implementasi *Speech Recognition*

Implementasi *speech recognition* ini menggunakan *HTML5 Speech Recognition API* yang mempunyai akses ke *stream audio*

browser dan mengubahnya kedalam teks. Bahasa yang dipakai pada *speech recognition* ini adalah Bahasa Indonesia. Kode sumber dari implementasi kasus penggunaan jenis makanan yang dapat dibagikan ditunjukkan pada Kode Sumber 8.1.

```
1. <script>
2.   function startDictation() {
3.
4.     if (window.hasOwnProperty('webkitSpeechRecogni
5.       tion')) {
6.       var recognition = new webkitSpeechRecogniti
7.         on();
8.         recognition.continuous = false;
9.         recognition.interimResults = false;
10.
11.        recognition.lang = "id-ID";
12.        recognition.start();
13.
14.        recognition.onresult = function(e) {
15.          document.getElementById('message-to-
16.            send').value = e.results[0][0].transcript;
17.          recognition.stop();
18.          document.getElementById('message-to-
19.            send').submit();
20.        };
21.
22.        recognition.onerror = function(e) {
23.          recognition.stop();
24.        }
25.      }
26.    }
27.  }
28. </script>
```

Kode Sumber 4.5 Implementasi Speech Recognition

4.5 Implementasi Kasus Penggunaan

Implementasi kasus penggunaan menggunakan Bahasa Python. Pada subbab ini akan menjelaskan dan menampilkan kode yang digunakan dalam pembuatan *chatbot* pada beberapa kasus penggunaan. Pada implementasi kasus penggunaan ini terdapat variabel regex sebagai *regular expression* yang menampung pola kalimat dari setiap kasus pengguna dan fungsi interpret untuk menentukan predikat atau relasi apa saja yang digunakan. Yang berbeda dari setiap kasus penggunaan adalah isi dari regex dan predikat. Daftar predikat dapat dilihat pada Kode Sumber 8.1.

4.5.1 Implementasi Kasus Penggunaan Jenis Makanan yang dapat Dibagikan

Kode sumber dari implementasi kasus penggunaan jenis makanan yang dapat dibagikan ditunjukkan pada Kode Sumber 8.1.

```

1. entity = Plus(Pos("NNP") | Pos("NN") | Pos("NNS")
2.     , "entity")
3.
4. regex = (Lemma("food") + Lemma("list") ) | (Lemma
5.     ("type") + Pos("IN") + entity ) | (Pos("WP") + \
6.     Pos("NNS") + Pos("MD") + Pos("VB") + Lemm
7.     a("share")) | (Lemma("type") + Pos("IN") + \
8.     entity + Lemma("that") + Pos("MD") + Pos
9.     ("VB") + Lemma("share"))
10.
11. def interpret(self, match):
12.     entity = HasKeyword("Gifood")
13.     label = ListOfFood(entity)
14.
15.     return label, "define"

```

Kode Sumber 4.6 Implementasi Kasus Penggunaan Jenis Makanan yang dapat Dibagikan

Variabel entity sebagai variabel hasil POS Tagging dari makanan. Predikat yang digunakan pada fungsi ini adalah: HasKeyword dan LisOfFood.

4.5.2 Implementasi Kasus Penggunaan Apa itu Gifood

Kode sumber dari implementasi kasus penggunaan apa itu Gifood ditunjukkan pada Kode Sumber 8.17.

```

1. regex = ( Lemma("what") + Lemma("be") + Thing() +
  \
2.   Question(Pos(".")) ) | Thing() + Pos(".") | P
  os("VBZ") + Thing()
3.
4. def interpret(self, match):
5.   label = DefinitionOf(match.thing)
6.   return label, "define"

```

Kode Sumber 4.7 Implementasi Kasus Penggunaan Apa itu Gifood

Predikat yang digunakan pada kasus ini adalah: DefinitionOf. 'match.thing' digunakan untuk mengambil nilai dari fungsi Thing() yang mengembalikan kata "Gifood".

4.5.3 Implementasi Kasus Penggunaan Cara Kerja Gifood

Kode sumber dari implementasi kasus penggunaan cara kerja Gifood ditunjukkan pada Kode Sumber 8.1.

```

1. regex = ( Lemma("how") + Thing() + ( Lemma("work"
  ) | Pos("VB") ) + Question(Pos(".")) ) | ( Lemma(
  "how") + Pos("VBZ") + \
2.   Thing() + Pos("VB") + Pos("NN"))
3.
4. def interpret(self, match):
5.   work = HowWorks(match.thing)
6.

```



```
7. return work, "define"
```

Kode Sumber 4.8 Implementasi Kasus Penggunaan Cara Kerja Gifood

Predikat yang digunakan pada kasus ini adalah : HowWorks. ‘match.thing’ digunakan untuk mengambil nilai dari fungsi Thing() yang mengembalikan kata “Gifood”.

4.5.4 Implementasi Kasus Penggunaan Contact Person Gifood

Kode sumber dari implementasi kasus penggunaan contact person Gifood ditunjukkan pada Kode Sumber 8.1.

```
1. regex = Lemma("contact") + Lemma("person") + Thing() | Lemma("contact") + Thing() | Lemma("email") + Thing() | \
2. Lemma("social") + Lemma("media") + Thing() | Lemma("how") + Pos("TO") + Pos("VB") + Thing()
3.
4. def interpret(self, match):
5.     label = CPGifood(match.thing)
6.
7.     return label, "define"
```

Kode Sumber 4.9 Implementasi Kasus Penggunaan Contact Person Gifood

Predikat yang digunakan pada kasus ini adalah : CPGifood. ‘match.thing’ digunakan untuk mengambil nilai dari fungsi Thing() yang mengembalikan kata “Gifood”.

4.5.5 Implementasi Kasus Penggunaan Mitra Gifood

Kode sumber dari implementasi kasus penggunaan mitra Gifood ditunjukkan pada Kode Sumber 8.1

```
1. regex = (Lemma("partner") + Pos("IN") + Thing() + Question(Pos("."))) | (Pos("NNS") | Pos("NNP") + \
```

```

2.         Thing())
3.
4. def interpret(self, match):
5.     mitra = HasKeyword("Mitra")
6.     label = PartnerGifood(mitra)
7.
8.     return label, "define"

```

Kode Sumber 4.10 Implementasi Kasus Penggunaan Mitra Gifood

Predikat yang digunakan pada kasus ini adalah : HasKeyword dan PartnerGifood.

4.5.6 Implementasi Kasus Penggunaan Cara Mengetahui Kelayakan Makanan

Kode sumber dari implementasi kasus penggunaan cara mengetahui kelayakan makanan ditunjukkan pada Kode Sumber 8.1.

```

1. entity = Group(Pos("NNP") | Pos("NN") | Pos("NNS"
2. ), "entity")
3. regex = Lemma("how") + Lemma("to") + Pos("VB") +
4.     entity + Lemma("eligibility") + Question(Pos("."))
5.
6. def interpret(self, match):
7.     layak = HasKeyword("Gifood")
8.     label = FoodEligibility(layak)
9.
10.    return label, "define"

```

Kode Sumber 4.11 Implementasi Kasus Penggunaan Cara Mengetahui Kelayakan Makanan

Variabel entity digunakan untuk menampung jenis kata “makanan”. Predikat yang digunakan pada kasus ini adalah : HasKeyword dan FoodEligibility.

4.5.7 Implementasi Kasus Penggunaan Apa itu Food Warrior

Kode sumber dari implementasi kasus penggunaan apa itu Food Warrior ditunjukkan pada Kode Sumber 8.1.

```

1. regex = ( Lemma("what") + Lemma("be") + Question(
   Pos("DT")) + \
2.     Thing() + Question(Pos(".")) ) | (Thing()
   )
3.
4. def interpret(self, match):
5.     label = DefinitionOf(match.thing)
6.     return label, "define"

```

Kode Sumber 4.12 Implementasi Kasus Penggunaan Apa itu Food Warrior

Predikat yang digunakan pada kasus ini adalah : DefinitionOf. ‘match.thing’ digunakan untuk mengambil nilai dari fungsi Thing() yang mengembalikan kata “Food Warrior”.

4.5.8 Implementasi Kasus Penggunaan Cara Menjadi Food Warrior

Kode sumber dari implementasi kasus penggunaan cara menjadi Food Warrior ditunjukkan pada Kode Sumber 8.1.

```

1. regex = Lemma("how") + Lemma("to") + Lemma("beco
   me") + Question(Pos("DT")) + \
2.     Thing() + Question(Pos("."))
3.
4. def interpret(self, match):
5.     label = HowToBe(match.thing)
6.
7.     return label, "define"

```

Kode Sumber 4.13 Implementasi Kasus Penggunaan Cara Menjadi Food Warrior

Predikat yang digunakan pada kasus ini adalah : HowToBe. ‘match.thing’ digunakan untuk mengambil nilai dari fungsi Thing() yang mengembalikan kata “Food Warrior”.

4.5.9 Implementasi Kasus Penggunaan Mencari Volunteer

Kode sumber dari implementasi kasus penggunaan mencari volunteer ditunjukkan pada Kode Sumber 8.1.

```

1. regex = Lemma("how") + Lemma("to") + Pos("VB") +
   Pos("DT") + Lemma("volunteer") + Question(Pos("."))
2.
3. def interpret(self, match):
4.     volunteer = HasKeyword("Food Warrior")
5.     label = HowWorks(volunteer)
6.
7.     return label, "define"

```

Kode Sumber 4.14 Implementasi Kasus Penggunaan Mencari Volunteer

Predikat yang digunakan pada kasus ini adalah : HasKeyword dan HowWorks.

4.5.10 Implementasi Kasus Penggunaan Makanan Berlebih

Kode sumber dari implementasi kasus penggunaan makanan berlebih ditunjukkan pada Kode Sumber 8.1.

```

1. entity = Group(Pos("NNP") | Pos("NN") | Pos("NNS")
   ), "entity")
2.
3. regex = Lemma("what") + Pos("IN") + Lemma("excessive") + entity + Question(Pos("."))
4.
5. def interpret(self, match):
6.     makanan = HasKeyword("Gifood")

```

```

7.     label = FoodExcessive(makanan)
8.
9.     return label, "define"

```

Kode Sumber 4.15 Implementasi Kasus Penggunaan Makanan Berlebih

Variabel entity digunakan untuk menampung jenis kata “makanan”. Predikat yang digunakan pada kasus ini adalah : HasKeyword dan FoodExcessive.

4.5.11 Implementasi Kasus Penggunaan Mencari Makanan yang Tersedia

Kode sumber dari implementasi kasus penggunaan mencari makanan yang tersedia ditunjukkan pada Kode Sumber 8.1 sampai dengan Kode Sumber 4.20. Pada implementasi kasus penggunaan ini terdapat variable regex sebagai *regular expression* yang menampung pola kalimat dari setiap kasus pengguna.

Pada kasus penggunaan ini terdapat beberapa fungsi dari *library* rdflib. Fungsi rdflib.Graph() untuk menginisiasi *triple* pada file RDF. Fungsi parse() untuk memuat file RDF yang akan digunakan. fungsi query() untuk melakukan proses query.

4.5.11.1 Pencarian Semua Makanan yang Tersedia

Kode sumber dari implementasi pencarian semua makanan yang tersedia ditunjukkan pada Kode Sumber 8.1.

```

1.  regex = Lemma("food") + Lemma("available")
2.
3.  def interpret(self, match):
4.
5.      date = date.today()
6.      today = str(date) + "T00:00:00"
7.
8.      g = rdflib.Graph()
9.      g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gif
    ood.rdf")
10.
11.     q = ""

```

```

12.         SELECT DISTINCT ?x0 ?p ?x1 ?name ?lok
asi WHERE {
13.         ?x0 rdfs:label "Transaksi"@en.
14.         ?x0 ta:MemilikiMakanan ?x2.
15.         ?x2 rdfs:label ?name.
16.         ?x0 ta:TerdapatDi ?x4.
17.         ?x4 rdfs:label ?lokasi.
18.         ?x0 ta:Status "ongoing"^^xsd:string.
19.         ?x0 ta:Expired_at ?date.
20.         FILTER(?date > '"" + today + '""'^^x
sd:dateTime).
21.         ?x0 ?p1 ?x1.
22.         ?p1 a owl:DatatypeProperty.
23.         ?p1 rdfs:label ?p.
24.         }""
25.     qres = g.query(q)

```

Kode Sumber 4.16 Implementasi Pencarian Semua Makanan yang Tersedia

Pada pencarian semua makanan yang tersedia dilakukan query untuk makanan yang memiliki status “*ongoing*” yang berarti masih tersedia dan memiliki batas tanggal pengambilan melebihi tanggal hari ini. Variable *today* digunakan untuk mendapatkan tanggal pada hari ini dan diubah dalam format *xsd:dateTime*.

4.5.11.2 Pencarian Makanan Berdasarkan Nama Makanan

Kode sumber dari implementasi pencarian makanan berdasarkan nama makanan ditunjukkan pada Kode Sumber 8.1.

```

1. regex = Pos("PRP") + Pos("VBP") + ( Thing() | Pos
("DT") + Thing() | Pos("TO") + Thing())
2.
3.     def interpret(self, match):
4.
5.         date = date.today()
6.         today = str(date) + "T00:00:00"
7.

```

```

8.         val = match.thing
9.         translations = translator.translate(val,
        src='en', dest='id')
10.        value = translations.text
11.
12.        g = rdflib.Graph()
13.        g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\
        \\gifood.rdf")
14.        q = """
15.            SELECT DISTINCT ?p ?x1 ?name ?lokasi
        WHERE {
16.                ?x0 ta:MemilikiMakanan ?x2.
17.                ?x2 rdfs:label ?name.
18.                ?x0 ta:TerdapatDi ?x4.
19.                ?x4 rdfs:label ?lokasi.
20.                FILTER regex(?name, '""'+value+ ""''
        , "i").
21.                ?x0 ta:Expired_at ?date.
22.                FILTER(?date > '""'+ today + ""'^^x
        sd:dateTime).
23.                ?x0 ?p1 ?x1.
24.                ?p1 a owl:DatatypeProperty.
25.                ?p1 rdfs:label ?p.
26.            }""""
27.
28.        qres = g.query(q)

```

Kode Sumber 4.17 Implementasi Pencarian Makanan Berdasarkan Nama Makanan

Pada pencarian makanan berdasarkan nama makanan hanya menampilkan makanan yang masih tersedia. Variabel `val` digunakan untuk mengambil hasil dari fungsi `Thing()`. Dikarenakan pada proses POS Tagging menggunakan Bahasa Inggris sedangkan data pada ontologi menggunakan Bahasa Indonesia, nilai yang didapat harus diubah kembali ke dalam Bahasa Indonesia.

Query dalam kasus ini menggunakan sintaks `"FILTER regex(?name, '""'+value+ ""''', "i")."`. Sintaks ini digunakan agar pengguna dapat mencari makanan hanya dari salah

satu kata dari nama makanan tersebut atau nama lengkap. Sehingga dapat memudahkan pengguna dalam mencari nama makanan yang mempunyai nama yang panjang.

4.5.11.3 Pencarian Makanan Berdasarkan Jumlah Porsi

Kode sumber dari implementasi pencarian makanan berdasarkan jumlah porsi ditunjukkan pada Kode Sumber 8.1.

```

1. regex = ( Pos("PRP") | Lemma("i") ) + Pos("VBP")
   + Lemma("food") + Quantity()
2.
3. def interpret(self, match):
4.     date = date.today()
5.     today = str(date) + "T00:00:00"
6.
7.     val = match.quantity
8.     val_min = str(int(val)-5)
9.     val_max = str(int(val)+5)
10.    value = str(val)
11.
12.    g = rdflib.Graph()
13.    g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gif
   ood.rdf")
14.    q = """
15.        SELECT DISTINCT ?p ?x1 ?name ?lokasi WHERE
   E {
16.            ?x0 ta:TerdapatDi ?x2.
17.            ?x2 rdfs:label ?lokasi.
18.            ?x0 ta:Jumlah ?jumlah.
19.            FILTER (?jumlah = "" + value + "" || (?
   jumlah >= "" + val_min + "" && ?jumlah <= "" + val
   _max + "")).
20.            ?x0 ta:MemilikiMakanan ?x3.
21.            ?x3 rdfs:label ?name.
22.            ?x0 ta:Expired_at ?date.
23.            FILTER(?date > "" + today + ""^^xsd:d
   ateTime).
24.            ?x0 ?p1 ?x1.
25.            ?p1 a owl:DatatypeProperty.

```



```

26.         ?p1 rdfs:label ?p.
27.         }""""
28.     qres = g.query(q)

```

Kode Sumber 4.18 Implementasi Pencarian Makanan Berdasarkan Jumlah Porsi

Pada pencarian makanan berdasarkan jumlah porsi hanya menampilkan makanan yang masih tersedia. Variabel `val` digunakan untuk mengambil hasil dari fungsi `Quantity()` yaitu jumlah yang diinginkan oleh pengguna. Selain memberikan daftar makanan sesuai jumlah yang diinginkan pengguna, sistem ini juga akan memberikan jumlah makanan sesuai rentang terdekat dari jumlah yang diinginkan. Rentang tersebut disimpan dalam variabel `val_min` dan `val_max`.

4.5.11.4 Pencarian Makanan Berdasarkan Kota

Kode sumber dari implementasi pencarian makanan berdasarkan kota ditunjukkan pada Kode Sumber 8.1.

```

1. regex = (Pos("DT") + Lemma("food") | Lemma("food"
2.         ) ) + (Pos("VBZ") + Pos("IN") | Pos("IN") )+ Thing()
3.
4. def interpret(self, match):
5.     date = date.today()
6.     today = str(date) + "T00:00:00"
7.
8.     val = match.thing
9.
10.    g = rdflib.Graph()
11.    g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gif
12.         ood.rdf")
13.    q = """
14.        SELECT DISTINCT ?p ?x1 ?name ?lokasi WHERE
15.        E {
16.            ?x0 ta:TerdapatDi ?x2.
17.            ?x2 rdfs:label ?lokasi.

```

```

15.     FILTER regex(?lokasi, '"" +value+ ""',
16.     "i").
17.     ?x0 ta:MemilikiMakanan ?x3.
18.     ?x3 rdfs:label ?name.
19.     ?x0 ta:Expired_at ?date.
20.     FILTER(?date > '"" + today + ""'^^xsd:d
21.     ateTime).
22.     ?x0 ?p1 ?x1.
23.     ?p1 a owl:DatatypeProperty.
24.     ?p1 rdfs:label ?p.
25.     }""
26.     qres = g.query(q)

```

Kode Sumber 4.19 Implementasi Pencarian Makanan Berdasarkan Kota

Pada pencarian makanan berdasarkan kota hanya menampilkan makanan yang masih tersedia. Variabel `val` digunakan untuk mengambil hasil dari fungsi `Thing()` yaitu nama kota. Nama kota disimpan dalam fungsi `Thing()` dikarenakan nama kota memiliki jenis sebagai kata benda.

4.5.11.5 Pencarian Makanan Berdasarkan Tanggal

Kode sumber dari implementasi pencarian makanan berdasarkan kota ditunjukkan pada Kode Sumber 8.1.

```

1.  regex = ( Pos("PRP") | Lemma("i") ) + Pos("VBP")
2.  + Lemma("food") + \
3.  (Lemma("on") | Lemma("before") | Lemma("a
4.  fter")) + Date()
5.  def interpret(self, match):
6.  date = date.today()
7.  today = str(date) + "T00:00:00"
8.  val = match.date
9.  value = str(val)
10. tgl = value + "T00:00:00"
11. if tgl == today :
12.     tgl = today

```

```

13.
14.     if Lemma("on") == "on" :
15.         tanda = "="
16.     elif Lemma("before") == "before" :
17.         tanda = "<"
18.     elif Lemma("after") == "after" :
19.         tanda = ">"
20.
21.     g = rdflib.Graph()
22.     g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gifo
ood.rdf")
23.     q = ""
24.     SELECT DISTINCT ?p ?x1 ?name ?lokasi WHERE
E {
25.         ?x0 ta:TerdapatDi ?x2.
26.         ?x2 rdfs:label ?lokasi.
27.         ?x0 ta:MemilikiMakanan ?x3.
28.         ?x3 rdfs:label ?name.
29.         ?x0 ta:Expired_at ?date.
30.         FILTER(?date "" + tanda + "" "" + tgl +
""'^^xsd:dateTime).
31.         ?x0 ?p1 ?x1.
32.         ?p1 a owl:DatatypeProperty.
33.         ?p1 rdfs:label ?p.
34.     }""
35.     qres = g.query(q)

```

Kode Sumber 4.20 Implementasi Pencarian Makanan Berdasarkan Tanggal

Pada pencarian makanan berdasarkan tanggal hanya menampilkan makanan yang masih tersedia. Variabel `val` digunakan untuk mengambil hasil dari fungsi `Date()`.

Pada kasus ini pengguna harus memasukkan format tanggal yang benar, yaitu YYYY-MM-DD. Pengguna dapat mencari makanan sebelum, sesudah atau pada tanggal yang diinginkan dan sesuai hari ini. Lemma("on") digunakan untuk pertanyaan pada tanggal sekian, Lemma("before") digunakan

untuk pertanyaan sebelum tanggal sekian, dan Lemma(“after”) digunakan untuk pertanyaan sesudah tanggal sekian.

4.5.12 Implementasi Kasus Penggunaan Berbagi makanan Berlebih

Kode sumber dari implementasi kasus penggunaan Berbagi makanan berlebih pada Kode Sumber 9.1. Kode sumber ini adalah untuk berbagi makanan ketika nama makanan yang ingin dibagikan tersedia pada website. Untuk penjelasan kode sumber ditunjukkan pada Tabel 9.1

4.5.12.1 Implementasi Menambahkan Nama Makanan Baru

Kode sumber dari implementasi untuk menambahkan nama makanan baru ditunjukkan pada Kode Sumber 9.1. Kode sumber ini adalah untuk berbagi makanan ketika nama makanan yang ingin dibagikan tersedia pada website. Untuk penjelasan kode sumber ditunjukkan pada Tabel 9.1

```

1. indv = eval(sys.argv[1])[0]
2. nama_makanan = eval(sys.argv[1])[1]
3.
4. g = Graph()
5. g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gifood.
   rdf")
6.
7. nm = URIRef(indv)
8.
9.
10. ind = URIRef("http://www.w3.org/2002/07/owl#Named
   Individual")
11. tipe_makanan = URIRef("http://www.semanticweb.org
   /asus/ontologies/2019/1/untitled-ontology-
   49#Makanan")
12.
13. g.add( (nm, RDF.type, tipe_makanan ) )
14.
15. g.add( (nm, RDFS.label, Literal(nama_makanan, lan
   g="en")) )
16. g.add( (nm, RDFS.label, Literal(nama_makanan, dat
   atype=XSD.string)) )

```

```

17.
18. g.serialize("C:\\xampp7\\htdocs\\TA\\dbpedia\\gif
    ood.rdf")

```

Kode Sumber 4.21 Implementasi Menambahkan Nama Makanan Baru

Tabel 4.2 Penjelasan Kode Sumber 9.1

No. Baris	Kegunaan
1, 2	<i>Input</i> dari form
4,5	Inisiasi untuk memuat file RDF
7	Variabel nama <i>instance</i>
10	Variabel untuk tipe individu
11	Variabel bahwa transaksi ini termasuk dalam kelas Makanan
13-16	Menambah <i>instance</i> dengan format S P O
18	Menyimpan kedalam file RDF

4.6 Implementasi Proses Mengubah Bahasa Alami Menjadi Query SPARQL

Implementasi kasus perhitungan menggunakan Bahasa PHP dari Moodle. Pada subbab ini akan menjelaskan dan menampilkan kode perhitungan *meaningful learning* yang digunakan dalam pembuatan *plugin meaningful learning* pada sisi aplikasi perangkat bergerak pada beberapa kasus penggunaan.

4.6.1 Implementasi Fungsi Main

Fungsi Main merupakan sebuah titik awal dan titik akhir dari program ini. Dalam fungsi main ini digunakan untuk membaca input dari pengguna serta proses – proses untuk mengawali jalannya program.

4.6.1.1 Implementasi Fungsi Membaca File RDF

Library yang digunakan untuk memproses file RDF pada fungsi ini adalah RdfLib. Berikut adalah kode yang digunakan untuk memuat file RDF yang digunakan dalam program ini.

```
1. g = Graph()
2. g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gifood.
   rdf")
```

Kode Sumber 4.22 Implementasi Fungsi Membaca File RDF

4.6.1.2 Implementasi Mengubah Bahasa Indonesia Menjadi Bahasa Inggris

Dalam proses POS Tagging, Natural Language Processing Toolkit (NLTK) hanya bisa menangani input dalam Bahasa Inggris. Sehingga input pengguna yang menggunakan Bahasa Indonesia diubah terlebih dahulu ke dalam Bahasa Inggris menggunakan API Google Translate dari Google.

```
1. from googletrans import Translator
2. translator = Translator()
3.
4.
5. for question in questions:
6.     translations = translator.translate(question,
   dest='en')
7.     pertanyaan = translations.text
```

Kode Sumber 4.23 Implementasi Mengubah Bahasa Indonesia Menjadi Bahasa Inggris

4.6.1.3 Implementasi Fungsi Query Sparql

Bahasa alami yang sudah diubah ke dalam bentuk query sparql akan diproses menggunakan fungsi yang ada pada RdfLib. Dan juga menampilkan hasil dari query tersebut.

```
1. sparqlquery = query
2. results = g.query(sparqlquery)
```

```

3. for row in resu [14]lts:
4.     d = str(row.asdict()['x1']).toPython()
5.     print (d)

```

Kode Sumber 4.24 Implementasi Fungsi Query Sparql

4.6.2 Implementasi Fungsi Tokenisasi dan POS Tagging

Langkah pertama dalam proses perubahan bahasa alami menjadi query adalah mengubah kalimat yang utuh kedalam token – token menggunakan fungsi tokenisasi yang telah disediakan oleh Natural Language Toolkit (NLTK). Setelah proses tokenisasi, masing – masing token yang di dapatkan akan di beri nilai atau ditentukan masing masing jenis katanya yang disebut dengan Pos Tagging. Berikut adalah kode implementasi dalam proses tokenisasi dan POS Tagging.

```

1. tokens = nltk.wordpunct_tokenize(string)
2. tags = nltk.pos_tag(tokens)

```

Kode Sumber 4.25 Implementasi Fungsi Tokenisasi dan POS Tagging

4.6.3 Implementasi Fungsi Mengubah Menjadi Query SPAQRL

Dalam fungsi ini, predikat yang sudah ditentukan pada saat pencocokkan pola kalimat akan disusun kedalam bentuk query sparq. Pada query ini akan ditampilkan prefix yang disimpan dalam variabel preamble, diikuti dengan format query sparql. Format query sparql adalah S P O, dimana subjek akan diisi dengan variable, kemudian predikat akan diisi dari hasil pada waktu peoncocokkan pola kalimat dan objek dapat diisi varibel atau sebuah string.

```

1. template = u"{preamble}\n" +\
2.           u"SELECT DISTINCT {select} WHERE {{\n"
           +\

```

```

3.         u"{expression}\n" +\
4.         u"}\n"
5. head = adapt(e.get_head())
6. if full:
7.     select = u"*"
8. else:
9.     select = head
10. y = 0
11. xs = []
12. for node in e.iter_nodes():
13.     for relation, dest in e.iter_edges(node):
14.         if relation is IsRelatedTo:
15.             relation = u"?y{}".format(y)
16.             y += 1
17.             xs.append(triple(adapt(node), relation, a
dapt(dest),
18.                             indentation=1))
19. sparql = template.format(preamble=settings.SPARQL
_PREAMBLE,
20.                           select=select,
21.                           expression=u"\n".join(xs
))
22. return select, sparql

```

Kode Sumber 4.26 Implementasi Fungsi Mengubah Menjadi Query SPAQRL

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk Tugas Akhir ini. Pengujian dilakukan pada kasus penggunaan dari sistem 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

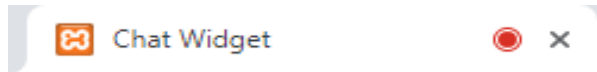
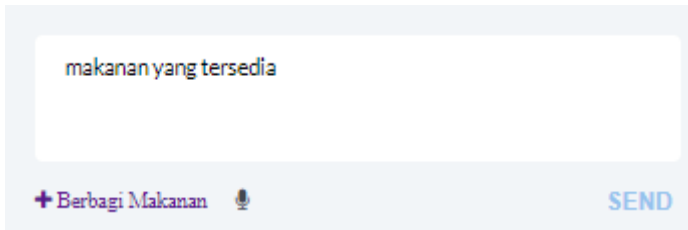
Perangkat	Spesifikasi
Perangkat Keras	Prosesor: Intel® Core™ i7-4720HQ CPU @ 2.60GHz 2.59GHz Memori: 8096MB RAM
Perangkat Lunak	Sistem Operasi: Windows 10 Pro 64-bit Perangkat Pengembang: Python 2.7.14, PHP 7.2.6 Basis Data: Ontologi

5.2 Pengujian *Speech Recognition*

Pengujian dilakukan dengan menekan tombol *microphone* yang tersedia pada bagian bawah *chatbot*. Apabila tombol *microphone* di tekan, akan muncul lingkaran merah pada tab di *browser*. Rincian skenario pengujian *speech recognition* ini ditunjukkan pada Tabel 5.2. Hasil yang diujikan ditunjukkan pada Gambar 5.1 dan Gambar 5.2.

Tabel 5.2 Pengujian *Speech Recognition*

Nama	Pengujian <i>speech recognition</i>
Tujuan Pengujian	Menguji fitur <i>speech recognition</i> .
Kondisi Awal	Pengguna berada pada halaman utama <i>chatbot</i> .
Data Uji	-
Langkah Pengujian	1. Pengguna menekan tombol <i>microphone</i> . 2. Pengguna berbicara dalam Bahasa Indonesia.
Hasil yang diharapkan	Sistem dapat menampilkan kalimat yang diucapkan oleh pengguna.
Hasil yang didapat	Kalimat pertanyaan sesuai dengan yang diucapkan oleh pengguna.
Hasil Pengujian	Berhasil
Kondisi Akhir	Kalimat yang diucapkan oleh pengguna berhasil ditampilkan.

**Gambar 5.1 Ketika Tombol Microphone Ditekan****Gambar 5.2 Hasil Pengujian *Speech Recognition***

5.3 Pengujian Fungsionalitas

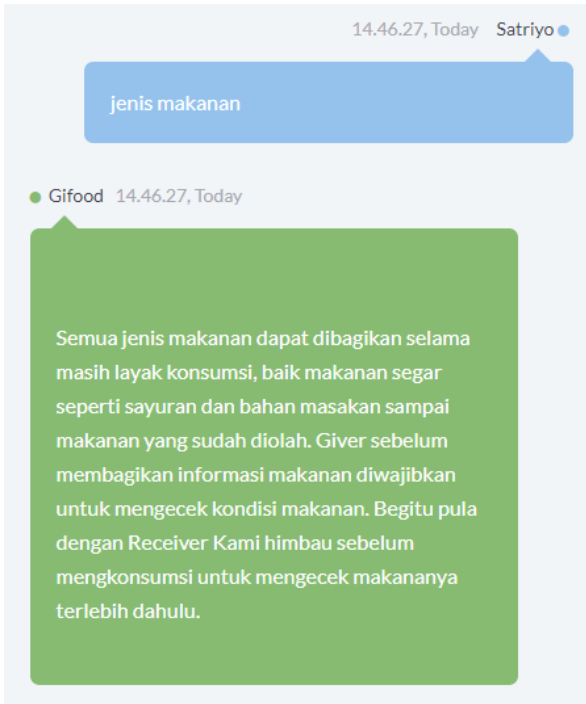
Pengujian yang dilakukan berdasarkan *black box testing*, dimana setiap fungsionalitas *chatbot* diuji apakah sesuai dengan hasil yang diharapkan. Pengujian dilakukan dengan memasukkan pertanyaan sesuai dengan fungsionalitas yang ada.

5.3.1 Pengujian Fungsionalitas Jenis Makanan yang Dapat di Bagikan.

Rincian skenario pengujian pada kasus penggunaan jenis makanan yang dapat dibagikan ditunjukkan pada Tabel 5.3. Hasil pengujian ditunjukkan pada Gambar 5.3.

Tabel 5.3 Pengujian Fungsionalitas Jenis Makanan yang Dapat di Bagikan.

No. Pengujian	UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian fungsionalitas jenis makanan yang dapat dibagikan
Tujuan Pengujian	Menguji fungsionalitas untuk menampilkan jenis makanan yang dapat dibagikan.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Jenis makanan.
Hasil yang diharapkan	Sistem menjawab jenis makanan apa yang dapat dibagikan.
Hasil yang didapat	Penjelasan tentang jenis makanan yang dapat dibagikan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang jenis makanan yang dapat dibagikan.



Gambar 5.3 Hasil Pengujian Fungsionalitas Jenis Makanan yang Dapat di Bagikan.

5.3.2 Pengujian Fungsionalitas Apa itu Gifood.

Rincian skenario pengujian pada kasus penggunaan apa itu Gifood ditunjukkan pada Tabel 5.4. Hasil pengujian ditunjukkan pada Gambar 5.4.

Tabel 5.4 Pengujian Fungsionalitas Apa itu Gifood.

No. Pengujian	UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian fungsionalitas apa itu Gifood.

Tujuan Pengujian	Menguji fungsionalitas untuk menampilkan pengertian dari Gifood.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Apa itu Gifood
Hasil yang diharapkan	Sistem menjawab pengertian tentang Gifood.
Hasil yang didapat	Penjelasan tentang pengertian Gifood.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab pertanyaan apa itu Gifood.



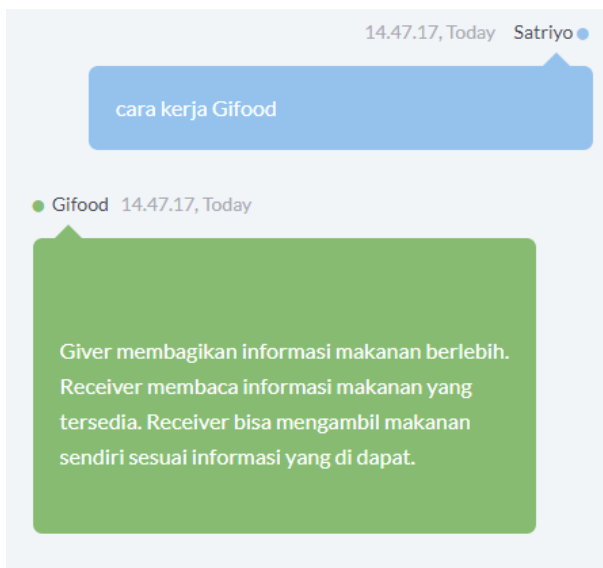
Gambar 5.4 Hasil Pengujian Fungsionalitas Apa itu Gifood.

5.3.3 Pengujian Fungsionalitas Cara Kerja Gifood

Rincian skenario pengujian pada kasus penggunaan cara kerja Gifood ditunjukkan pada Tabel 5.5. Hasil pengujian ditunjukkan pada Gambar 5.5.

Tabel 5.5 Pengujian Fungsionalitas Cara Kerja Gifood

No. Pengujian	UC-003
Referensi Kasus Penggunaan	UC-003
Nama	Pengujian fungsionalitas cara kerja Gifood
Tujuan Pengujian	Menguji fungsionalitas untuk menampilkan cara kerja Gifood.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Cara kerja Gifood.
Hasil yang diharapkan	Sistem menjawab cara kerja Gifood.
Hasil yang didapat	Penjelasan tentang alur cara kerja Gifood.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang cara kerja Gifood.



Gambar 5.5 Hasil Pengujian Fungsionalitas Cara Kerja Gifood

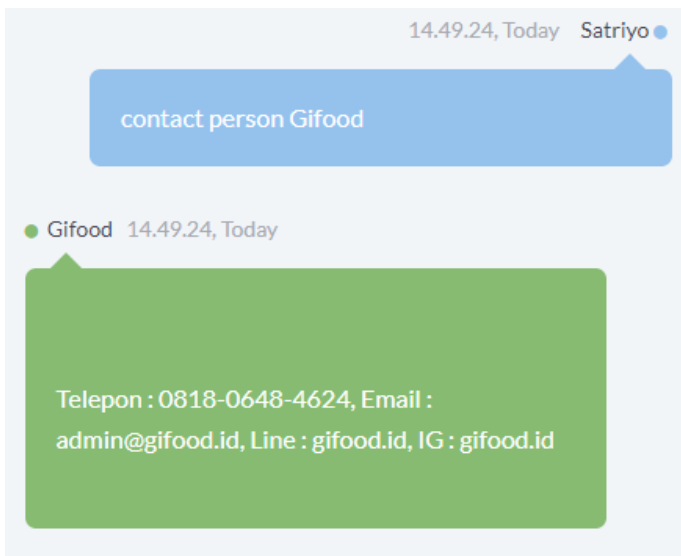
5.3.4 Pengujian Fungsionalitas Contact Person Gifood.

Rincian skenario pengujian pada kasus penggunaan contact person Gifood ditunjukkan pada Tabel 5.6. Hasil pengujian ditunjukkan pada Gambar 5.6.

Tabel 5.6 Pengujian Fungsionalitas Contact Person Gifood.

No. Pengujian	UC-004
Referensi Kasus Penggunaan	UC-004
Nama	Pengujian fungsionalitas contact person Gifood.
Tujuan Pengujian	Menguji fungsionalitas untuk menampilkan semua contact person dari Gifood .
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-

Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Contact person Gifood.
Hasil yang diharapkan	Sistem menjawab contact person Gifood.
Hasil yang didapat	Daftar contact person dari Gifood.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menampilkan contact person Gifood.



Gambar 5.6 Hasil Pengujian Fungsionalitas Contact Person Gifood.

5.3.5 Pengujian Fungsionalitas Mitra Gifood.

Rincian skenario pengujian pada kasus penggunaan mitra Gifood ditunjukkan pada Tabel 5.7. Hasil pengujian ditunjukkan pada Gambar 5.7.

Tabel 5.7 Pengujian Fungsionalitas Mitra Gifood.

No. Pengujian	UC-005
Referensi Kasus Penggunaan	UC-005
Nama	Pengujian fungsionalitas mitra Gifood.
Tujuan Pengujian	Menguji fungsionalitas untuk menampilkan mitra Gifood.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Mitra Gifood.
Hasil yang diharapkan	Sistem menampilkan mitra dari Gifood.
Hasil yang didapat	Penjelasan tentang mitra Gifood.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang mitra Gifood.



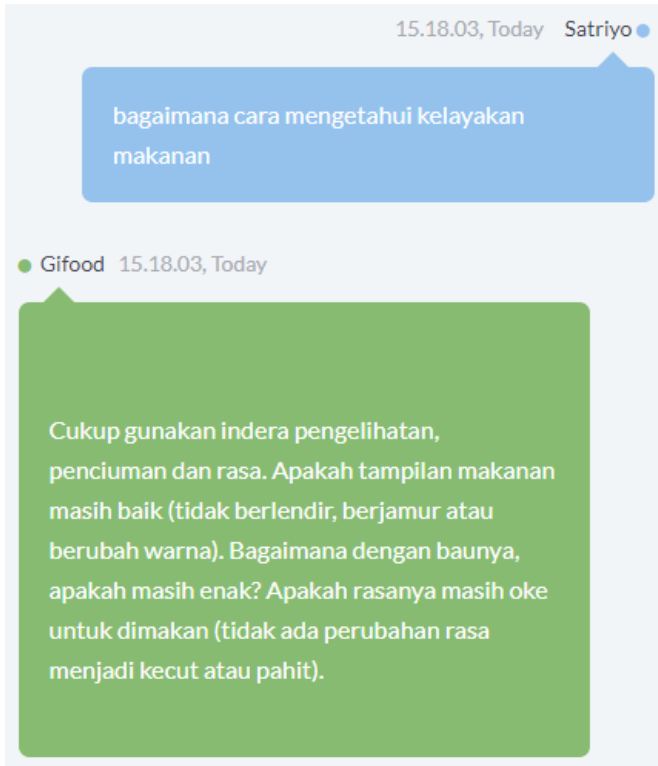
Gambar 5.7 Hasil Pengujian Fungsionalitas Mitra Gifood.

5.3.6 Pengujian Fungsionalitas Cara Mengetahui Kelayakan Makanan.

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.8. Hasil pengujian ditunjukkan pada Gambar 5.8.

Tabel 5.8 Pengujian Fungsionalitas Cara Mengetahui Kelayakan Makanan.

No. Pengujian	UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian fungsionalitas cara mengetahui kelayakan makanan.
Tujuan Pengujian	Menguji fungsionalitas untuk mengetahui kelayakan makanan.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Bagaimana cara mengetahui kelayakan makanan.
Hasil yang diharapkan	Sistem menjawab cara mengetahui kelayakan makanan.
Hasil yang didapat	Penjelasan tentang cara mengetahui kelayakan makanan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang cara mengetahui kelayakan makanan.



Gambar 5.8 Hasil Pengujian Fungsionalitas Cara Mengetahui Kelayakan Makanan.

5.3.7 Pengujian Fungsionalitas Apa itu Food Warrior.

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.9. Hasil pengujian ditunjukkan pada Gambar 5.9.

Tabel 5.9 Pengujian Fungsionalitas Apa itu Food Warrior.

No. Pengujian	UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian fungsionalitas apa itu Food Warrior.
Tujuan Pengujian	Menguji fungsionalitas untuk menjelaskan apa itu Food Warrior.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Apa itu Food Warrior.
Hasil yang diharapkan	Sistem menjawab apa itu Food Warrior.
Hasil yang didapat	Penjelasan tentang apa itu Food Warrior.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang apa itu Food Warrior.



Gambar 5.9 Hasil Pengujian Fungsionalitas Apa itu Food Warrior.

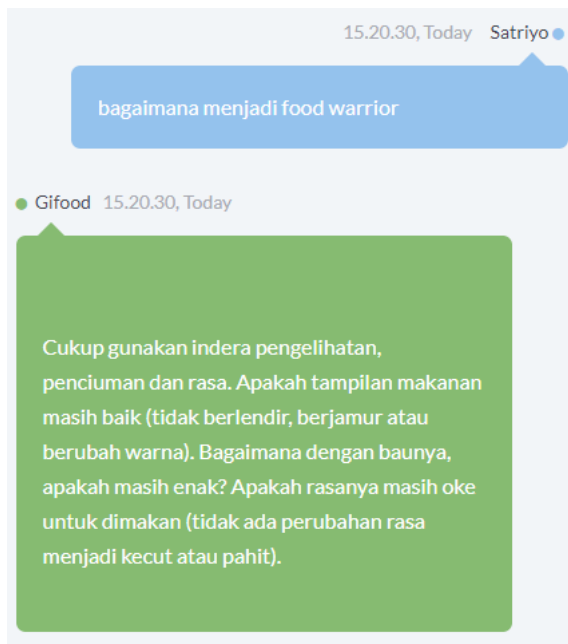
5.3.8 Pengujian Fungsionalitas Cara Menjadi Food Warrior.

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.10. Hasil pengujian ditunjukkan pada Gambar 5.10.

Tabel 5.10 Pengujian Fungsionalitas Cara Menjadi Food Warrior.

No. Pengujian	UC-008
Referensi Kasus Penggunaan	UC-008
Nama	Pengujian fungsionalitas cara menjadi Food Warrior
Tujuan Pengujian	Menguji fungsionalitas untuk menjelaskan bagaimana cara menjadi Food Warrior.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message"

	2. Pengguna menekan tombol “Send”
Pertanyaan	Bagaimana cara menjadi Food Warrior
Hasil yang diharapkan	Sistem menjelaskan bagaimana cara menjadi Food Warrior.
Hasil yang didapat	Penjelasan tentang bagaimana cara mejadi Food Warrior.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang cara menjadi Food Warrior.



Gambar 5.10 Hasil Pengujian Fungsionalitas Cara Menjadi Food Warrior.

5.3.9 Pengujian Fungsionalitas Mencari Volunteer.

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.11. Hasil pengujian ditunjukkan pada Gambar 5.11.

Tabel 5.11 Pengujian Fungsionalitas Mencari Volunteer.

No. Pengujian	UC-009
Referensi Kasus Penggunaan	UC-009
Nama	Pengujian fungsionalitas mencari volunteer.
Tujuan Pengujian	Menguji fungsionalitas bagaimana cara mencari seorang volunteer.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Bagaimana mencari seorang volunteer
Hasil yang diharapkan	Sistem menjawab cara mencari seorang volunteer.
Hasil yang didapat	Penjelasan tentang cara mencari seorang volunteer.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang cara mencari seorang volunteer.



Gambar 5.11 Hasil Pengujian Fungsionalitas Mencari Volunteer.

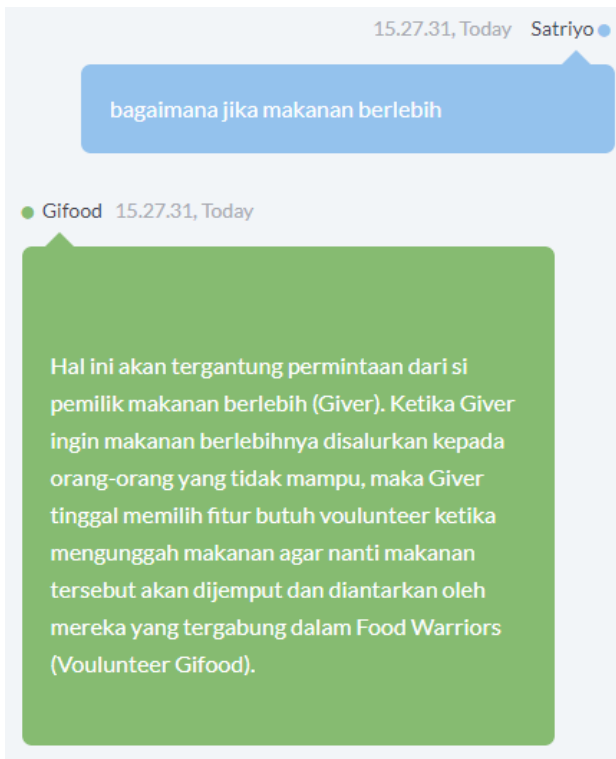
5.3.10 Pengujian Fungsionalitas Makanan Berlebih.

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.12. Hasil pengujian ditunjukkan pada Gambar 5.12.

Tabel 5.12 Pengujian Fungsionalitas Makanan Berlebih.

No. Pengujian	UC-010
Referensi Kasus Penggunaan	UC-010
Nama	Pengujian fungsionalitas makanan berlebih.
Tujuan Pengujian	Menguji fungsionalitas untuk menjelaskan bagaimana jika makanan yang dibagikan berlebih.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"

Pertanyaan	Bagaimana jika makanan berlebih.
Hasil yang diharapkan	Sistem menjawab bagaimana jika makanan berlebih.
Hasil yang didapat	Penjelasan tentang bagaimana jika makanan yang dibagikan berlebih.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menjawab tentang bagaimana jika makanan berlebih.



Gambar 5.12 Hasil Pengujian Fungsionalitas Makanan Berlebih.

5.3.11 Pengujian Fungsionalitas Mencari Makanan yang Tersedia dengan Nama, Lokasi, Waktu, Status Ketersediaan, dan Jumlah Porsi.

Dalam pengujian fungsionalitas ini, akan dilakukan pengujian untuk masing masing parameter yang akan dijelaskan pada poin berikutnya.

5.3.11.1 Pengujian Fungsionalitas Mencari Semua Makanan yang Tersedia

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.13. Hasil pengujian ditunjukkan pada Gambar 5.13.

Tabel 5.13 Pengujian Fungsionalitas Mencari Semua Makanan yang Tersedia

No. Pengujian	UC-011
Referensi Kasus Penggunaan	UC-011
Nama	Pengujian fungsionalitas mencari semua makanan yang tersedia.
Tujuan Pengujian	Menguji fungsionalitas mencari semua makanan yang tersedia.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Makanan yang tersedia
Hasil yang diharapkan	Sistem menampilkan daftar makanan yang tersedia.
Hasil yang didapat	Daftar makanan yang tersedia.
Hasil Pengujian	Berhasil

Kondisi Akhir	Sistem dapat menampilkan daftar makanan yang tersedia.
---------------	--



Gambar 5.13 Hasil Pengujian Fungsionalitas Mencari Semua Makanan yang Tersedia

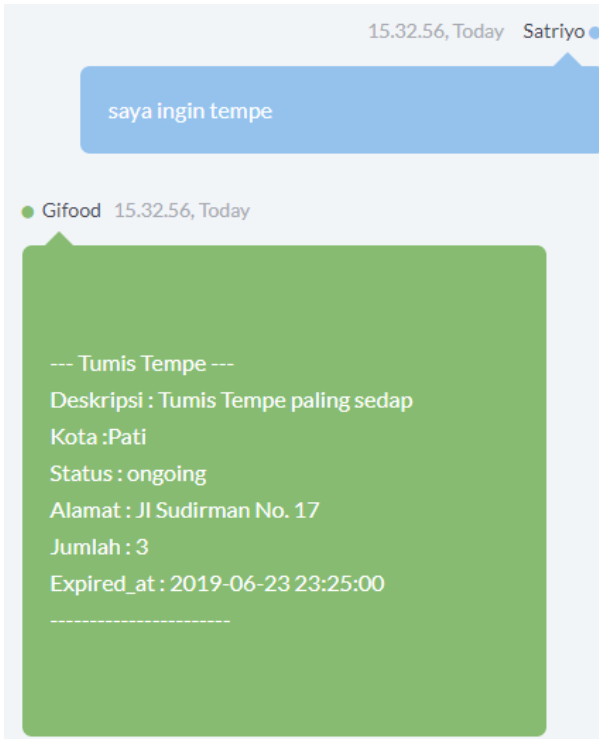
5.3.11.2 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Nama Makanan

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.14. Hasil pengujian ditunjukkan pada Gambar 5.14.

Tabel 5.14 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Nama Makanan

No. Pengujian	UC-011
---------------	--------

Referensi Kasus Penggunaan	UC-011
Nama	Pengujian fungsionalitas mencari makanan berdasarkan nama makanan.
Tujuan Pengujian	Menguji fungsionalitas mencari makanan berdasarkan nama makanan.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Saya ingin tempe
Hasil yang diharapkan	Sistem menampilkan daftar makanan yang terdapat nama tempe.
Hasil yang didapat	Daftar makanan yang memiliki nama tempe dan terdapat nama tempe.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menampilkan daftar makanan berdasarkan nama makanan.



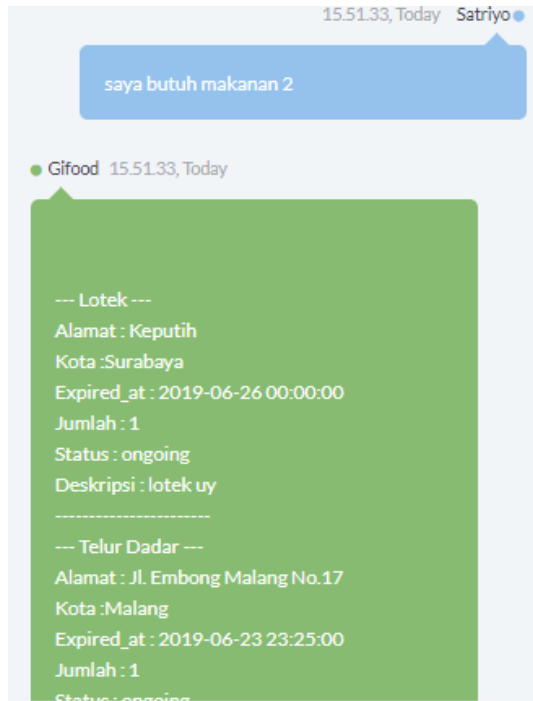
Gambar 5.14 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Nama Makanan

5.3.11.3 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Jumlah Porsi

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.15. Hasil pengujian ditunjukkan pada Gambar 5.15.

Tabel 5.15 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Jumlah Porsi

No. Pengujian	UC-011
Referensi Kasus Penggunaan	UC-011
Nama	Pengujian fungsionalitas mencari makanan berdasarkan jumlah porsi.
Tujuan Pengujian	Menguji fungsionalitas mencari makanan berdasarkan jumlah porsi.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Saya butuh makanan 2.
Hasil yang diharapkan	Sistem menampilkan daftar makanan dengan jumlah porsi 2 atau menyarankan daftar makanan dengan jumlah porsi yang mendekati 2.
Hasil yang didapat	Daftar makanan dengan jumlah porsi 2 atau makanan dengan jumlah porsi yang mendekati 2.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menampilkan daftar makanan berdasarkan jumlah porsi.



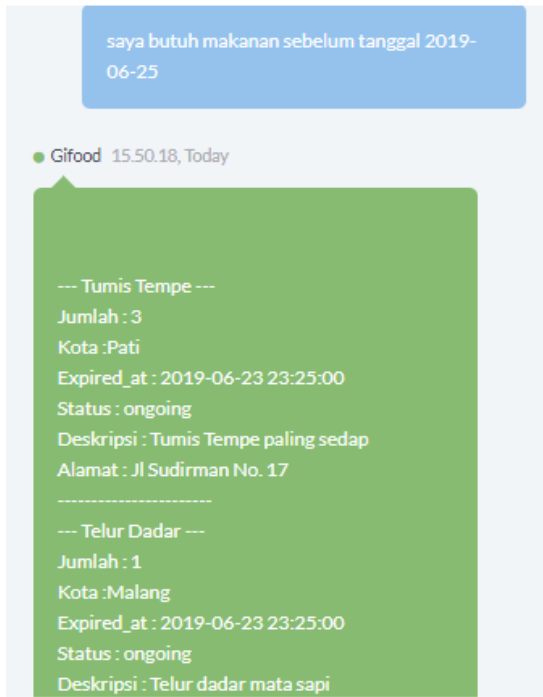
Gambar 5.15 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Jumlah Porsi

5.3.11.4 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Tanggal

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.16. Hasil pengujian ditunjukkan pada Gambar 5.16.

Tabel 5.16 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Tanggal

No. Pengujian	UC-011
Referensi Kasus Penggunaan	UC-011
Nama	Pengujian fungsionalitas mencari makanan berdasarkan tanggal.
Tujuan Pengujian	Menguji fungsionalitas mencari makanan berdasarkan tanggal.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 2. Pengguna menekan tombol "Send"
Pertanyaan	Saya butuh makanan sebelum tanggal 2019-06-25
Hasil yang diharapkan	Sistem menampilkan daftar makanan sebelum tanggal yang diinginkan.
Hasil yang didapat	Daftar makanan yang tersedia sebelum tanggal yang diinginkan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menampilkan daftar makanan yang tersedia berdasarkan tanggal.



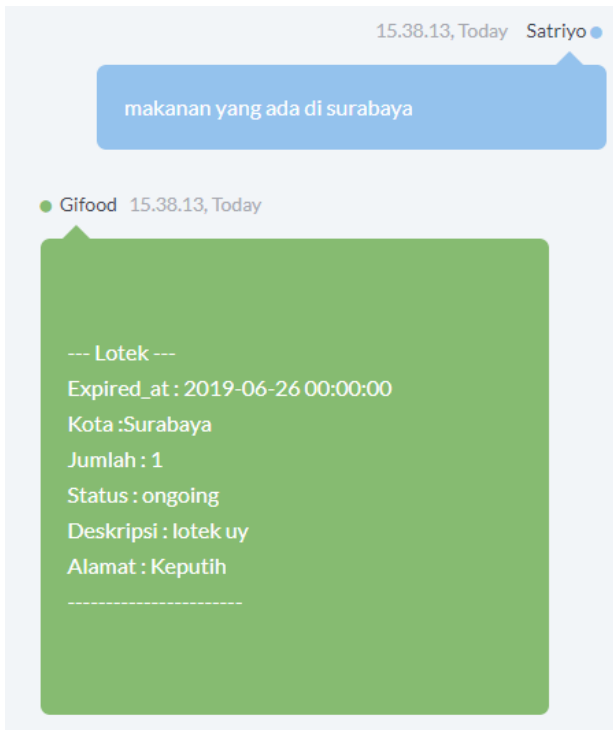
Gambar 5.16 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Tanggal

5.3.11.5 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Lokasi

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.17. Hasil pengujian ditunjukkan pada Gambar 5.17.

Tabel 5.17 Pengujian Fungsionalitas Mencari Makanan Berdasarkan Lokasi

No. Pengujian	UC-011
Referensi Kasus Penggunaan	UC-011
Nama	Pengujian fungsionalitas mencari makanan berdasarkan lokasi.
Tujuan Pengujian	Menguji fungsionalitas mencari makanan berdasarkan lokasi.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 3. Pengguna memasukkan pertanyaan pada kolom pesan "Type your message" 4. Pengguna menekan tombol "Send"
Pertanyaan	Makanan yang ada di Surabaya
Hasil yang diharapkan	Sistem menampilkan daftar makanan sesuai lokasi yang diinginkan.
Hasil yang didapat	Daftar makanan yang tersedia sesuai lokasi yang diinginkan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Sistem dapat menampilkan daftar makanan yang tersedia berdasarkan lokasi.



Gambar 5.17 Hasil Pengujian Fungsionalitas Mencari Makanan Berdasarkan Lokasi

5.3.12 Pengujian Fungsionalitas Berbagi Makanan Berlebih.

Pada pengujian ini, pengguna mengisi form berbagi makanan. Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.18. Hasil pengujian ditunjukkan pada Gambar 5.18 sampai Gambar 5.20.

Tabel 5.18 Pengujian Fungsionalitas Berbagi Makanan Berlebih.

No. Pengujian	UC-012
Referensi Kasus Penggunaan	UC-012
Nama	Pengujian fungsionalitas berbagi makanan berlebih.
Tujuan Pengujian	Menguji fungsionalitas untuk berbagi makanan berlebih.
Kondisi Awal	Pengguna berada pada halaman utama chatbot.
Data Uji	-
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Berbagi Makanan” 2. Pengguna mengisi form yang disediakan 3. Pengguna menekan tombol “Submit”
Hasil yang diharapkan	Data makanan yang kita bagikan tersimpan pada ontologi.
Hasil yang didapat	Sukses menyimpan data makanan yang ingin dibagikan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Data makanan yang ingin dibagikan tersimpan.

Bagikan Makanan

Saya punya makanan untuk dibagikan

Jenis Makanan

Makanan ▼

Ingin menambahkan nama makanan baru ?

Ndas Manyung

Ndas Manyung Juwana

2

Jl. Panglima Sudirman No. 17

Nama Kota

Pati ▼

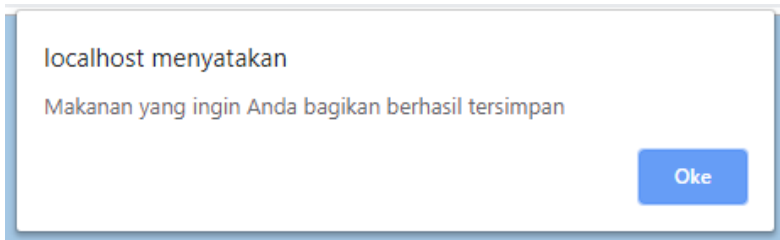
Dapat diambil sebelum tanggal

29/06/2019

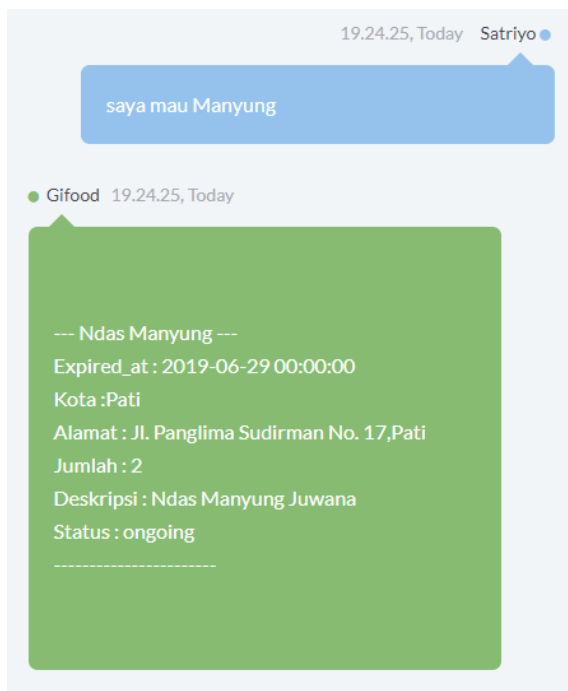
Submit

Back to Chatbot

Gambar 5.18 Pengisian Form Berbagi Makanan



Gambar 5.19 Notifikasi Bahwa Makanan Yang Diinginkan Berhasil



Gambar 5.20 Hasil Pengujian Fungsionalitas Berbagi Makanan Berlebih.

5.4 Pengujian *Natural Language Processing*

Pada pengujian *Natural Language Processing* (NLP) ini dilakukan dengan mencoba berbagai variasi pertanyaan sesuai kebutuhan fungsionalitas sistem. Pengujian kali ini dikatakan berhasil ketika *Natural Language Processing* (NLP) berhasil mengubah pertanyaan menjadi query sparql.

Pada pengukuran kinerja dari *Natural Language Processing* ini 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

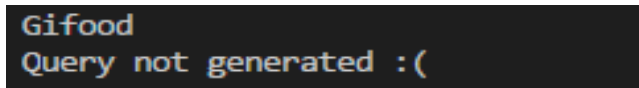
Dalam pengujian ini, format pertanyaan minimal terdiri dari dua jenis kata. Variasi pertanyaan yang digunakan pada pengujian ini serta hasil pengujian terdapat pada Tabel 12.1. Dari perhitungan dari pengujian ini didapatkan akurasi 97,75%. Berikut adalah perhitungan pengujian kinerja menggunakan *confusion matrix*:

$$Akurasi = \frac{50 + 35}{50 + 35 + 0 + 2} \times 100\%$$

$$Akurasi = 97,70\%$$

Pada pengujian ini, terdapat 87 kasus dimana 52 kasus benar dan 35 kasus salah. Pengujian kasus benar menggunakan variasi dari fungsionalitas sistem yang ada sedangkan pengujian kasus salah menggunakan variasi pertanyaan dengan format satu kata atau frasa yang tidak bisa ditangani oleh chatbot dan variasi pertanyaan selain kebutuhan fungsionalitas chatbot. Dari hasil pengujian didapatkan bahwa Chatbot Gifood berhasil mendeteksi 50 kasus benar dan mendeteksi 35 kasus salah. Chatbot tidak bisa mendeteksi 2 kasus dikarenakan pada pertanyaan “ada bakso?” dan “daftar food warrior” dideteksi sebagai satu frasa. Menurut Batasan masalah, format pertanyaan tidak boleh satu frasa.

Berikut adalah hasil dari pengujian *Natural Language Processing* (NLP) menggunakan satu kata. Pengujian dilakukan dengan menuliskan pertanyaan “Gifood”. Hasil yang didapatkan dari pengujian tersebut adalah sebagai berikut :



```
Gifood
Query not generated :(
```

Gambar 5.21 Hasil Pengujian dengan Satu Kata

Dari Gambar 5.21 menunjukkan bahwa *Natural Language Processing* (NLP) tidak dapat memahami atau mengubah pertanyaan menjadi query sparql.

Berikut adalah salah satu hasil dari pengujian *Natural Language Processing* (NLP) sesuai kebutuhan fungsionalitas sistem. Pengujian dilakukan dengan menuliskan pertanyaan “Gifood”. Hasil yang didapatkan dari pengujian tersebut adalah sebagai berikut :

```

Food Warrior ?

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

SELECT DISTINCT ?x1 WHERE {
  ?x0 rdfs:label "Food Warrior"@en.
  ?x0 ta:About ?x1.
}

Volunteer untuk membagikan makanan - makanan.

```

Gambar 5.22 Hasil Pengujian *Natural Language Processing*

Dari Gambar 5.22 menunjukkan bahwa *Natural Language Processing* (NLP) dapat memahami atau mengubah pertanyaan menjadi query sparql.

5.5 Evaluasi Pengujian Fungsionalitas

Berdasarkan pengujian yang telah dilakukan, semua pengujian fungsionalitas memberikan hasil yang sesuai dengan skenario yang direncanakan. Rangkuman mengenai hasil pengujian fungsionalitas yang diujicobakan ditunjukkan pada Tabel 5.19.

Tabel 5.19 Evaluasi Pengujian Fungsionalitas

Kode	Fungsionalitas	Terpenuhi
F01	Jenis makanan yang dapat di bagikan.	✓
F02	Apa itu Gifood.	✓
F03	Cara kerja Gifood	✓
F04	Contact Person Gifood	✓
F05	Mitra Gifood	✓
F06	Cara mengetahui kelayakan makanan	✓
F07	Apa itu Food Warrior	✓
F08	Cara menjadi Food Warrior	✓
F09	Mencari Volunteer	✓
F10	Makanan berlebih	✓
F11	Mencari makanan yang tersedia dengan nama, lokasi, waktu, status ketersediaan, dan jumlah porsi.	✓
F12	Berbagi makanan berlebih	✓

Berdasarkan data pada Tabel 5.1920, seluruh skenario pengujian berhasil dilakukan. Sehingga dapat disimpulkan bahwa fungsionalitas dari sistem *chatbot* bekerja sesuai dengan yang diharapkan.

5.6 Hasil Ujicoba Pengguna

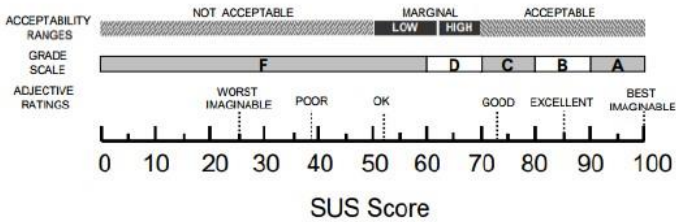
Selain melakukan ujicoba fungsional, ujicoba kepada pengguna digunakan untuk mendapatkan respon dari orang lain yang mencoba menggunakan chatbot ini. Dalam pengujian ini dilakukan dengan menggunakan *System Usability Scale* (SUS). SUS ini merupakan salah satu alat pengujian usability yang paling populer. SUS dikembangkan oleh John Brooke pada tahun 1986. SUS ini merupakan skala usability yang handal, populer, efektif dan murah. SUS memiliki 10 pertanyaan dan 5 pilihan jawaban. Pilihan jawaban terdiri dari sangat tidak setuju sampai sangat setuju. SUS memiliki skor minimal 0 dan skor maksimal 100. Pertanyaan dari SUS dapat dilihat pada Tabel 10.1.

Dalam cara menggunakan System Usability Scale (SUS) ada beberapa aturan dalam perhitungan skor SUS. Berikut ini aturan-aturan saat perhitungan skor pada kuesionernya:

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1.
2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
3. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

Hasil perhitungan dari SUS ditunjukkan pada Tabel 5.21.

[16]



Gambar 5.23 Skala Perhitungan SUS

Responden yang melakukan pengujian sistem ini adalah orang yang sudah pernah memakai aplikasi perpesanan atau pernah memakai *chatbot*. Selain itu responden telah diberitahu fungsionalitas pada sistem ini.

Tabel 5.20 Hasil Ujicoba Pengguna

Pertanyaan	Responden 1	Responden 2	Responden 3
Q1	4	4	3
Q2	3	3	3
Q3	4	4	4
Q4	3	2	2
Q5	4	4	4
Q6	3	3	3
Q7	3	3	3
Q8	3	3	3
Q9	3	3	3
Q10	3	3	3
Jumlah	33	32	31
Nilai (jumlah x 2.5)	83	80	78
Skor Rata-Rata (Hasil Akhir)			80.33

Dari hasil perhitungan pada Tabel 5.21, hasil ujicoba pengguna didapatkan skor akhir adalah 80.33. Berdasarkan skala pada Gambar 5.23 didapatkan bahwa sistem ini mendapatkan kategori baik dengan nilai B dan masuk dalam kategori diterima.

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

Kesimpulan dari Tugas Akhir ini adalah implementasi *Natural Language Processing* pada *chatbot* telah berhasil dibangun dengan solusi dari beberapa rumusan masalah sebagai berikut:

1. Implementasi *chatbot* berbasis website. Implementasi dilakukan dengan membuat tampilan antarmuka dengan pengguna menggunakan bahasa pemrograman HTML, PHP, dan Javascript.
2. Implementasi *speech recognition* pada *chatbot* dibangun dengan menggunakan *HTML5 Speech Recognition API* menggunakan bahasa Javascript.
3. *Natural Language Processing* berhasil di implementasikan untuk mengubah bahasa alami yaitu pertanyaan pengguna ke dalam bahasa query sparql. Pengolahan bahasa alami ini menggunakan beberapa proses, antara lain: tokenisasi, *POS Tagging*, pencocokan kalimat menggunakan persamaan ekspresi reguler yang di bentuk menggunakan jenis kata. Dalam proses query sparql digunakan *library rdflib*. *Rdflib* digunakan untuk memuat atau membaca file ontologi RDF, memproses untuk mendapatkan hasil sesuai query yang telah dibentuk dari proses NLP, dan digunakan untuk membuat *instance* baru untuk menyimpan data transaksi dan makanan baru yang belum pernah dibagikan sebelumnya.
4. Pengujian *Natural Language Processing* dilakukan dengan berbagai variasi pertanyaan dan didapatkan akurasi 97,75%. Pengujian *chatbot* dilakukan dengan *black box testing*, dimana

setiap fungsionalitas *chatbot* telah terpenuhi. Kemudian, *chatbot* diuji dengan uji coba pengguna dan dinilai berdasarkan skor *System Usability Scale* (SUS) dimana *chatbot* mendapatkan nilai B (kategori diterima).

6.2 Saran

Dari kesimpulan yang penulis buat pada sub-bab sebelumnya, dan berdasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan, penulis memberikan saran :

1. *Chatbot* lebih berinteraksi dengan memberikan saran kepada pengguna apabila hasil yang diinginkan tidak ditemukan pada database chatbot.
2. Memperbanyak variasi pola pertanyaan.
3. Mengintegrasikan sistem yang sudah dibuat dengan sistem Gifood.id.

DAFTAR PUSTAKA

- [1] A. Nurfikry, "Chatbot, Tren Baru untuk Pengalaman Perbankan Lebih Mudah," Kata.ai, 20 Agustus 2018. [Online]. Available: <https://blog.kata.ai/id/post/chatbot-tren-baru-untuk-pengalaman-perbankan-lebih-mudah/>. [Accessed 27 Desember 2018].
- [2] A. Nurfikry, "Mengapa Bisnis Anda Memerlukan Chatbot?," Kata.ai, 17 Juli 2018. [Online]. Available: <https://blog.kata.ai/id/post/mengapa-bisnis-anda-memerlukan-chatbot/>. [Accessed 27 Desember 2018].
- [3] Gifood, "Gifood," Gifood, [Online]. Available: <https://gifood.id/>. [Accessed 27 Desember 2018].
- [4] S. A. Abdul-Kader and D. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 6, pp. 73-74, 2015.
- [5] C. J. Baby, F. A. Khan and S. J. N., "Home Automation using IoT and a Chatbot using Natural Language Processing," *International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017]*, 2017.
- [6] Derwin Suhartono, S.Kom., MTI., "Natural Language Processing," School of Computer Science Binus, [Online]. Available: <http://socs.binus.ac.id/2013/06/22/natural-language-processing/>. [Accessed 27 Desember 2018].
- [7] R. Posevkin and I. Bessmertny, "Translation of natural language queries to structured data sources," *9th International Conference on Application of Information and Communication Technologies (AICT)*, 2015.
- [8] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit*, 2009.

- [9] "Stemming and lemmatization," Cambridge University Press, [Online]. Available: <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>. [Accessed 24 June 2019].
- [10] F. Rashed, A. Luthfi, A. Dinakaramani and R. Manurung, "Building an Indonesian Rule-Based Part-of-Speech Tagger," *International Conference on Asian Language Processing (IALP)*, 2014.
- [11] M.A.Anusuya and S.K.Katti, "Speech Recognition by Machine: A Review," (*IJCSIS*) *International Journal of Computer Science and Information Security*, vol. 6, p. 181, 10 October 2009.
- [12] Machinalis, "Welcome to Quepy's documentation!," 17 November 2017. [Online]. Available: <https://quepy.readthedocs.io/en/latest/index.html>. [Accessed 24 June 2019].
- [13] E. K. Nurnawati, "Representasi Database Berbasis Ontologi Dengan Resource Description Framework (RDF)," *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III*, 2012.
- [14] RDFlib, "RDFlib," [Online]. Available: <https://rdflib.readthedocs.io/en/stable/>. [Accessed 15 6 2019].
- [15] Y. F. Badron, F. Agus and H. R. Hatta, "STUDI TENTANG PEMODELAN ONTOLOGI WEB SEMANTIK DAN PROSPEK PENERAPAN PADA BIBLIOGRAFI ARTIKEL JURNAL ILMIAH," *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, 2017.
- [16] Z. Sharfina and H. B. Santoso, "An Indonesian adaptation of the System Usability Scale (SUS)," *International Conference on Advanced Computer Science and Information Systems*, 2017.

LAMPIRAN A

Implementasi Fungsi Predikat

```
1. HasKeyword.relation = "rdfs:label"
2. HasKeyword.language = "en"
3.
4.
5. class HowToBe(FixedRelation):
6.     relation = "ta:HowToBe"
7.     reverse = True
8.
9. class DefinitionOf(FixedRelation):
10.    relation = "ta>About"
11.    reverse = True
12.
13. class RelationOfFood(FixedRelation):
14.    relation = "ta:MempunyaiJenisMakanan"
15.    reverse = True
16.
17. class ListOfFood(FixedRelation):
18.    relation = "ta:JenisMakanan"
19.    reverse = True
20.
21. class PartnerGifood(FixedRelation):
22.    relation = "ta:Mitra"
23.    reverse = True
24.
25. class HowWorks(FixedRelation):
26.    relation = "ta:HowItWorks"
27.    reverse = True
28.
29. class FoodEligibility(FixedRelation):
30.    relation = "ta:KelayakanMakanan"
31.    reverse = True
32.
33. class FoodExcessive(FixedRelation):
34.    relation = "ta:MakananBerlebih"
35.    reverse = True
36.
37. class CPGifood(FixedRelation):
38.    relation = "ta:ContactPerson"
39.    reverse = True
```

```
40.  
41. class Transaction(FixedRelation):  
42.     relation = "rdfs:comment"  
43.     reverse = True  
44.  
45. class ClassTransaction(FixedType):  
46.     fixedtype = "ta:Transaksi"  
47.  
48. class isStatus(HasDataProperty1):  
49.     datatype1 = "ongoing"
```

Kode Sumber 8.1 Implementasi Predikat

LAMPIRAN B

Implementasi Kasus Penggunaan Berbagai Makanan Berlebih

```
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. import datetime
9.
10. datetime.datetime.now().time()
11. datetime.time(15, 8, 24, 78915)
12.
13. time = str(datetime.datetime.now().time())
14.
15.
16. inp_makanan = eval(sys.argv[1])[0]
17. nama_makanan = eval(sys.argv[1])[1]
18. deskripsi_makanan = eval(sys.argv[1])[2]
19. jumlah_porsi = eval(sys.argv[1])[3]
20. alamat = eval(sys.argv[1])[4]
21. kota = eval(sys.argv[1])[5]
22. tanggal = eval(sys.argv[1])[6]
23.
24. RDF.type
25. RDF.datatype
26. RDFS.label
27. RDFS.comment
28. RDFS.seeAlso
29.
30. OWL.About
31. # = rdflib.term.URIRef(u'http://www.w3.org/1999/0
    2/22-rdf-syntax-ns#type')
32.
33. FOAF.knows
34. # = rdflib.term.URIRef(u'http://xmlns.com/foaf/0.
    1/knows')
35.
36.
```

```
37. g = Graph()
38. g.parse("C:\\xampp7\\htdocs\\TA\\dbpedia\\gifood.
    rdf")
39.
40. nm = URIRef("http://www.semanticweb.org/asus/onto
    logies/2019/1/untitled-ontology-
    49#Transaksi"+tanggal+time)
41.
42.
43. ind = URIRef("http://www.w3.org/2002/07/owl#Named
    Individual")
44. tipe = URIRef("http://www.semanticweb.org/asus/on
    tologies/2019/1/untitled-ontology-
    49#FoodOntology")
45. trans = URIRef("http://www.semanticweb.org/asus/o
    ntologies/2019/1/untitled-ontology-
    49#Transaksi")
46.
47.
48. relasi_makanan = URIRef("http://www.semanticweb.o
    rg/asus/ontologies/2019/1/untitled-ontology-
    49#MemilikiMakanan")
49. relasi_kota = URIRef("http://www.semanticweb.org/
    asus/ontologies/2019/1/untitled-ontology-
    49#TerdapatDi")
50.
51. expired_at = URIRef("http://www.semanticweb.org/a
    sus/ontologies/2019/1/untitled-ontology-
    49#Expired_at")
52. jumlah = URIRef("http://www.semanticweb.org/asus/
    ontologies/2019/1/untitled-ontology-49#Jumlah")
53. deskripsi = URIRef("http://www.semanticweb.org/as
    us/ontologies/2019/1/untitled-ontology-
    49#Deskripsi")
54. status = URIRef("http://www.semanticweb.org/asus/
    ontologies/2019/1/untitled-ontology-49#Status")
55. almt = URIRef("http://www.semanticweb.org/asus/on
    tologies/2019/1/untitled-ontology-49#Alamat")
56.
57. makanan = URIRef(inp_makanan)
58. kota = URIRef("http://www.semanticweb.org/asus/on
    tologies/2019/1/untitled-ontology-49#+kota)
59.
```

```
60.
61. g.add( (nm, RDF.type, tipe ) )
62. g.add( (nm, RDF.type, trans ) )
63.
64. g.add( (nm, expired_at, Literal(tanggal, datatype
=XSD.dateTime)) )
65. g.add( (nm, deskripsi, Literal(deskripsi_makanan,
datatype=XSD.string)) )
66. g.add( (nm, almt, Literal(alamat, datatype=XSD.st
ring)) )
67. g.add( (nm, jumlah, Literal(jumlah_porsi, datatyp
e=XSD.string)) )
68. g.add( (nm, status, Literal("ongoing", datatype=X
SD.string)) )
69.
70.
71. g.add( (nm, RDF.type, ind) )
72. g.add( (nm, relasi_makanan, makanan) )
73.
74. g.add( (nm, relasi_kota, kota) )
75. g.add( (nm, RDFS.label, Literal('Transaksi', lang
="en")) )
76. g.add( (nm, RDFS.label, Literal('Transaksi', data
type=XSD.string)) )
77.
78. g.serialize("C:\\xampp7\\htdocs\\TA\\dbpedia\\gif
ood.rdf")
```

Kode Sumber 9.1 Implementasi Kasus Penggunaan Berbagi Makanan Berlebih

Tabel 9.1 Penjelasan Kode Sumber 9.1

No. Baris	Kegunaan
10, 11	Mendapatkan waktu saat ini
16 - 22	<i>Input</i> dari form
24 – 33	Inisiasi fungsi yang ingin digunakan
37 -38	Inisiasi untuk memuat file RDF
40	Variabel nama transaksi. Format nama transaksi adalah Transaksi+Tanggal+Waktu
43	Variabel untuk tipe individu
44	Variabel bahwa transaksi ini termasuk dalam kelas FoodOntology
45	Variabel bahwa transaksi ini termasuk dalam kelas Transaksi
48	Relasi MemilikiMakanan
49	Relasi TerdapatDi
51	<i>Data properties</i> batas tanggal pengambilan
52	<i>Data properties</i> jumlah porsi
53	<i>Data properties</i> deskripsi
54	<i>Data properties</i> status
55	<i>Data properties</i> alamat
57	Variabel untuk nama makanan
58	Variabel untuk nama kota
61 – 76	Menambah <i>instance</i> dengan format S P O
78	Menyimpan kedalam file RDF

LAMPIRAN C

Daftar Pos Tagging
Tabel 10.1 Tabel Part-of-Speech

Tag	Deskripsi
CC	conjunction, coordinating
CD	cardinal number
DT	determiner
EX	existential there
FW	foreign word
IN	conjunction, subordinating or preposition
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list item marker
MD	verb, modal auxillary
NN	noun, singular or mass
NNS	noun, plural
NNP	noun, proper singular
NNPS	noun, proper plural
PDT	predeterminer
POS	possessive ending
PRP	pronoun, personal
PRP\$	pronoun, possessive
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	adverb, particle
SYM	symbol
TO	infinitival to
UH	interjection
VB	verb, base form
VBZ	verb, 3rd person singular present
VBP	verb, non-3rd person singular present
VBD	verb, past tense
VBN	verb, past participle
VBG	verb, gerund or present participle

Tag	Deskripsi
WDT	<i>wh</i> -determiner
WP	<i>wh</i> -pronoun, personal
WP\$	<i>wh</i> -pronoun, possessive
WRB	<i>wh</i> -adverb
.	punctuation mark, sentence closer
,	punctuation mark, comma
:	punctuation mark, colon
(contextual separator, left paren
)	contextual separator, right paren
NP	noun phrase
PP	prepositional phrase
VP	verb phrase
ADVP	adverb phrase
ADJP	adjective phrase
SBAR	subordinating conjunction
PRT	particle
INTJ	interjection

LAMPIRAN D

Kuisisioner Pengujian Chatbot
Tabel 11.1 Kuesioner Pengujian Terhadap Chatbot

No	Pertanyaan	STS	TS	RG	S	SS
1	Saya berpikir akan menggunakan sistem ini lagi					
2	Saya merasa sistem ini rumit untuk digunakan					
3	Saya merasa sistem ini mudah digunakan					
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini					
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya					
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)					
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat					
8	Saya merasa sistem ini membingungkan					
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini					
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini					

Keterangan :

Jawaban	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Ragu – Ragu	3
Setuju	4
Sangat Setuju	5

[Halaman ini sengaja dikosongkan]

LAMPIRAN E

Skenario dan Hasil Pengujian *Natural Language Processing*
Tabel 12.1 Tabel Skenario dan Hasil Pengujian *True Positive* untuk
Natural Language Processing

Pengujian <i>True Positive</i>				
No	Kode	Pertanyaan	Dugaaan	Hasil Pengujian
1	F01	makanan apa saja yang dapat dibagikan	Benar	Benar
2		jenis makanan	Benar	Benar
3		jenis makanan yang dapat dibagikan	Benar	Benar
4	F02	Apa itu gifood	Benar	Benar
5		Apa Gifood	Benar	Benar
6		apakah gifood	Benar	Benar
7	F03	cara kerja gifood	Benar	Benar
8		bagaimana gifood bekerja	Benar	Benar
9		bagaimana cara gifood membagikan makanan	Benar	Benar
10	F06	bagaimana cara mengetahui kelayakan makanan	Benar	Benar
11		ciri - ciri makanan yang layak	Benar	Benar
12		menguji kelayakan makanan	Benar	Benar
13	F07	Apa itu food warrior	Benar	Benar
14		apa food warrior	Benar	Benar
15		apakah food warrior	Benar	Benar
16	F08	aku pengen menjadi food warrior	Benar	Benar
17		cara menjadi food warrior	Benar	Benar
18		cara mendaftar food warrior	Benar	Benar
19		daftar food warrior	Benar	Salah
20	F10	bagaimana jika makanan berlebih	Benar	Benar
21		bagaimana jika makananku sisa	Benar	Benar
22		bagaimana jika makananku tidak habis	Benar	Benar

Pengujian <i>True Positive</i>				
No	Kode	Pertanyaan	Dugaaan	Hasil Pengujian
23		bagaimana jika makananku tidak ada yang mengambil	Benar	Benar
24	F04	contact person gifood	Benar	Benar
25		cara menghubungi gifood	Benar	Benar
26		contact gifood	Benar	Benar
27		sosial media gifood	Benar	Benar
28		mitra gifood	Benar	Benar
29	F05	apa saja mitra gifood	Benar	Benar
30		mitra usaha gifood	Benar	Benar
31		partner gifood	Benar	Benar
32		makanan yang ada	Benar	Benar
33	F11	makanan yang tersedia	Benar	Benar
34		semua makanan yang ada di gifood	Benar	Benar
35		apa saja makanan yang ada di gifood	Benar	Benar
36		tampilkan semua makanan	Benar	Benar
37		aku pengen bakso	Benar	Benar
38		ada bakso ?	Benar	Salah
39		apakah ada bakso ?	Benar	Benar
40		makanan yang ada di pati	Benar	Benar
41		makanan di sekitar pati	Benar	Benar
42		makanan di pati	Benar	Benar
43		saya butuh makanan sebelum tanggal 20 Juli 2019	Benar	Benar
44		saya butuh makanan pada tanggal 20 Juli 2019	Benar	Benar
45		saya butuh makanan setelah tanggal 20 Juli 2019	Benar	Benar
46		adakah makanan pada tanggal 20 Juli 2019	Benar	Benar
47		10 makanan	Benar	Benar
48		aku butuh 10 makanan	Benar	Benar
49		apakah ada makanan sebanyak 10 porsi	Benar	Benar

Pengujian <i>True Positive</i>				
No	Kode	Pertanyaan	Dugaaan	Hasil Pengujian
50	F12	saya butuh volunteeer	Benar	Benar
51		bagaimana mencari volunteeer	Benar	Benar
52		apakah ada volunteeer untuk membagikan makanan	Benar	Benar

Tabel 12.2 Tabel Skenario dan Hasil Pengujian *True Negative* untuk *Natural Language Processing*

Pengujian <i>True Negative</i>			
No	Pertanyaan	Dugaan	Hasil Pengujian
1	Bakso	Salah	Salah
2	Gifood	Salah	Salah
3	Halo	Salah	Salah
4	About Chatbot	Salah	Salah
5	10	Salah	Salah
6	Harga makanan	Salah	Salah
7	Makanan layak	Salah	Salah
8	Makanan di lokasi saya	Salah	Salah
9	Berbagi makanan	Salah	Salah
10	Selamat pagi	Salah	Salah
11	Apakah ada ongkos kirim	Salah	Salah
12	Gifood.id	Salah	Salah
13	Food Warriior	Salah	Salah
14	Faq	Salah	Salah
15	Apa yang membedakan makanan sehat dan yang tidak?	Salah	Salah
16	Apa beberapa kesamaan antara makanan sehat dan makanan cepat saji	Salah	Salah
17	Apakah ada makanan sehat dan makanan tidak sehat ?	Salah	Salah
18	Apa makanan sehat yang murah yang bisa dimakan setiap hari?	Salah	Salah
19	Mengapa makanan sehat tergolong mahal ?	Salah	Salah

Pengujian <i>True Negative</i>			
No	Pertanyaan	Dugaan	Hasil Pengujian
20	Berapa harga bakso	Salah	Salah
21	Bahan makanan	Salah	Salah
22	Warna makanan	Salah	Salah
23	Rasa makanan	Salah	Salah
24	Tanggal kadaluarsa	Salah	Salah
25	Tanggal pengambilan	Salah	Salah
26	Alamat pengambilan makanan	Salah	Salah
27	Makanan ringan	Salah	Salah
28	Makanan berat	Salah	Salah
29	Cemilan	Salah	Salah
30	Minuman	Salah	Salah
31	Makanan	Salah	Salah
32	Makanan sehat	Salah	Salah
33	Daftar anggota	Salah	Salah
34	Hidangan pembuka	Salah	Salah
35	Hidangan penutup	Salah	Salah

INDEX

A

akurasi · 93, 98

C

chatbot · vii, viii, ix, x, 1, 2, 3, 4, 5, 9, 10, 11, 12, 19, 20, 21, 23, 29, 31, 32, 39, 44, 62, 63, 64, 66, 67, 68, 70, 72, 74, 75, 77, 78, 80, 82, 84, 86, 88, 90, 94, 95, 96, 97, 98, 99, 110, 115

G

Gifood · vii, viii, ix, xi, 2, 3, 10, 11, 17, 19, 20, 21, 22, 23, 24, 25, 31, 38, 43, 44, 45, 46, 64, 65, 66, 67, 68, 69, 70, 93, 94, 99

N

Natural Language Processing · vii, viii, ix, x, 1, 2, 3, 4, 9, 11, 12, 13, 15, 16, 19, 20, 21, 22, 28, 58, 92, 93, 94, 98

P

POS Tag · 14, 16, 20, 22, 34, 44, 51, 58, 59, 98

O

ontologi · vii, viii, 4, 10, 16, 17, 18, 28, 29, 30, 51, 61, 89, 98

S

Speech Recognition · vii, viii, ix, x, 2, 3, 4, 5, 11, 14, 19, 20, 21, 31, 41, 42, 61, 62, 98
System Usability Scale – 95, 99

T

tokenisasi · 14, 16, 20, 21, 22, 33, 59, 98

[Halaman ini sengaja dikosongkan]

DAFTAR ISTILAH

- Chatbot* : Sebuah program komputer yang dirancang untuk menyimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara suara maupun teks.
- Variabel : Lokasi penyimpanan dan terkait nama simbolis yang berisi beberapa kuantitas yang diketahui atau tidak diketahui atau informasi, nilai.
- Query : Semacam kemampuan untuk menampilkan suatu data dari database dimana mengambil dari table-table yang ada di database, namun tabel tersebut tidak semua ditampilkan sesuai dengan yang kita inginkan.
- Relasi : Menghubungkan antara subjek dengan objek.
- Data Properties* : Menghubungkan individu dengan data literal (mis., String, angka, tipe data, dll.).
- Object Properties* : Menghubungkan individu dengan individu lain.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Satriyo Nugroho, si bungsu dari tiga bersaudara lahir di Pati pada tanggal 17 Januari 1997. Penulis telah menempuh pendidikan formal mulai dari SD N Karanglegi 02, SMP Negeri 3 Pati, SMA Negeri 1 Pati dan terakhir sebagai mahasiswa Departemen Informatika Institut Teknologi Sepuluh Nopember dengan rumpun mata kuliah Manajemen Informasi (2015-2019).

Selama perkuliahan, penulis aktif dalam organisasi kemahasiswaan, antara lain sebagai Staff Minat dan Bakat Himpunan Mahasiswa Teknik Computer-Informatika ITS 2016-2017, Staff Acara FTIF Festival 2016, Staff IFC ITS 2016, Fasilitator Acara Gerigi ITS 2016, Staff Reeva Schematics 2016, dan Ketua Schematics 2017.

Selama kuliah di Departemen Informatika ITS, penulis mengambil bidang minat Manajemen Informasi (MI) dengan ketertarikan penulis terdapat pada analisis data dan *website*. Penulis dapat dihubungi melalui surel satriyo.nug97@gmail.com.