

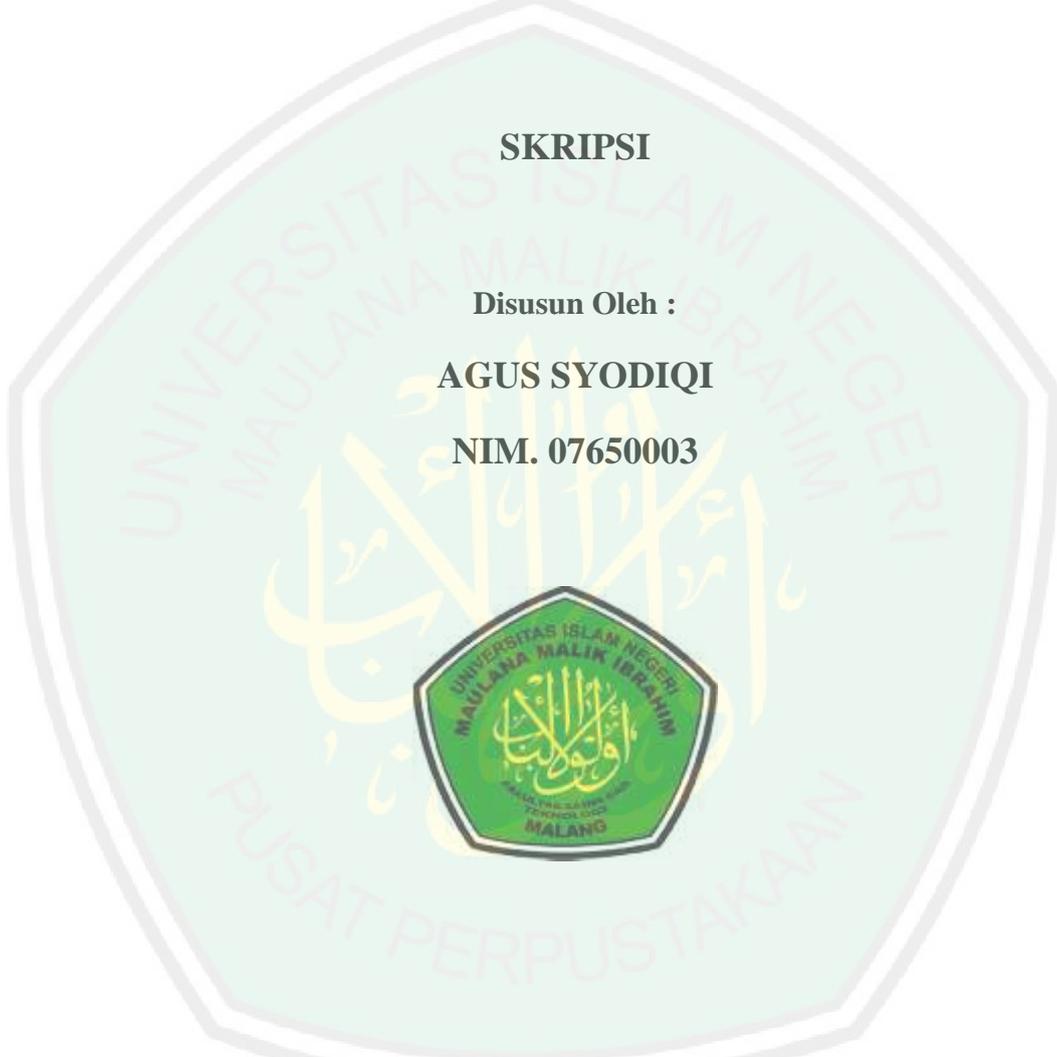
**APLIKASI REKOMENDASI PERGURUAN TINGGI
BERBASIS *SEMANTIC WEB* DENGAN METODE *MULTI
CRITERIA DECISION MAKING (MCDM)***

SKRIPSI

Disusun Oleh :

AGUS SYODIQI

NIM. 07650003



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK
IBRAHIM
MALANG
2013**

**APLIKASI REKOMENDASI PERGURUAN TINGGI
BERBASIS *SEMANTIC WEB* DENGAN METODE *MULTI
CRITERIA DECISION MAKING* (MCDM)**

SKRIPSI

Diajukan Kepada :

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang

Untuk Memenuhi Salah Satu Persyaratan Dalam

Memperoleh Gelar Sarjana S.Kom

Oleh :

AGUS SYODIQI

NIM. 07650003/S-1

JURUSAN TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM

MALANG

2013

**APLIKASI REKOMENDASI PERGURUAN TINGGI
BERBASIS *SEMANTIC WEB* DENGAN METODE *MULTI
CRITERIA DECISION MAKING***

SKRIPSI

Oleh :

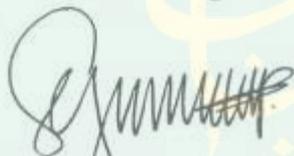
AGUS SYODIQI

NIM. 07650003

Telah Diperiksa dan Disetujui untuk Diuji :

Tanggal: 28 Oktober 2013

Pembimbing I



A'la Syaqui, M.Kom
NIP. 197712012008011007

Pembimbing II



Zainal Abidin, M.Kom
NIP. 197606132005011004

Mengetahui,

Ketua Jurusan Teknik Informatika



Dr. Cahyo Crysdian
NIP. 197404242009011008

**APLIKASI REKOMENDASI PERGURUAN TINGGI
BERBASIS *SEMANTIC WEB* DENGAN METODE *MULTI
CRITERIA DECISION MAKING (MCDM)***

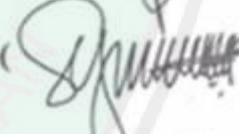
SKRIPSI

Oleh :
AGUS SYODIQI
NIM. 07650003

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal : 28 November 2013

Susunan Dewan Penguji :

- | | | Tanda Tangan |
|-----------------------|--|---|
| 1. Penguji Utama | : Dr. Muhammad Faisal, M.T
NIP. 19740510 20051 1007 | () |
| 2. Ketua Penguji | : Yunifa Miftachul Arif, MT
NIP. 19830616 201101 1004 | () |
| 3. Sekretaris Penguji | : A'la Syauqi, M.Kom
NIP. 19771 201200801 1007 | () |
| 4. Anggota Penguji | : Zainal Abidin, M.Kom
NIP. 19760613 200501 1004 | () |

Mengesahkan,
Ketua Jurusan Teknik Informatika



Dr. Cahyo Crvsdian
NIP. 19740424 200901 1008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Agus Syodiqi

NIM : 07650003

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau fikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Oktober 2013

Yang membuat pernyataan,

Agus Syodiqi
NIM. 07650003

MOTTO

إِلَّا مَا جَعَلْتَهُ سَهْلًا وَأَنْتَ تَجْعَلُ اللَّهُمَّ لَا سَهْلَ

الْحَزْنَ إِذَا شِئْتَ سَهْلًا

*“Ya Allah, tidak ada kemudahan kecuali yang Engkau buat mudah.
Dan engkau menjadikan kesedihan (kesulitan), jika Engkau
kehendaki pasti akan menjadi mudah”*

HR. Anas Bin Malik

Takut akan kegagalan seharusnya tidak menjadi alasan
untuk tidak mencoba sesuatu.

Rintangannya di depan kita tidak pernah sebesar
kekuatan yang ada di belakang kita

Semua yang kualami adalah rencana Allah SWT yang terbaik,

HALAMAN PERSEMBAHAN



“Sesungguhnya bersama kesulitan ada kemudahan”

QS. Al-Insyirah : 6

*“Yang Maha Pengasih lagi Maha Penyayang Allah SWT
“Terima kasih Ya Allah, Karena rencana-Mu Indah Bagiku. Semoga Kau slalu
limpahkan berkah-Mu padaku.” serta Sholawat atas Nabi Besar kita Muhammad
saw, Ya Allah, semoga setiap langkah selalu Engkau ridhoi dengan segala rahmat-Mu .*

*“Tugas kita bukanlah untuk berhasil. Tugas kita adalah untuk mencoba, karena di
dalam mencoba itulah kita menemukan dan belajar membangun kesempatan untuk
berhasil.”*

Saya Persembahkan karya ini kepada semua pihak yang telah membantu
dalam menyelesaikan karya ini

... dari seorang anak, untuk kedua orang tuanya...

Ayahanda Muhammad Shafwan (Alm.) & Umi Hidayati

*“Yang selalu ada dan percaya. Untuk cinta dan semua yang telah kalian berikan
sampai hari ini dan hari-hari berikutnya. Terima kasih atas kasih sayang kalian. Aku
merasa beruntung dan bahagia karena Allah telah menitipkanku pada kalian.”*

... dari seorang adik, untuk kakak-kakaknya ...

**Kanda Samsul Huda, S.Ag, Musrifatun, S.Pd.i, Istiqamah, S.Pd.i, Lilis Fitriani,
Mukhlis Alhadi, S.Pd.i dan Muhammad Ikhwan, S.H.i**

*“Makasih atas semua dukungan dan perhatian kalian slama ini, tanpa kalian
aku bukan aku yang sekarang.”*

... dari seorang kakak, untuk adik-adiknya

**Adinda Raodah Purnama Wati, Ahmad Shalehan
& Irhamul Khatim**

... dari seorang anak, untuk teman-teman seperjuangannya ...

Terima kasih

Atas segalanya,
semoga Allah SWT selalu melindungi dan menyayangi kita semua...

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Tiada kata yang pantas terucap selain Alhamdulillah sebagai wujud rasa syukur kepada Allah SWT, yang mana hanya dengan rahmat, taufik, hidayah, dan inayah-Nya. Penulis telah dapat menyelesaikan karya ilmiah dengan judul “*APLIKASI REKOMENDASI PERGURUAN TINGGI BERBASIS SEMANTIC WEB DENGAN METODE MULTI CRITERIA DECISION MAKING (MCDM)*” ini sebagai syarat untuk menyelesaikan pendidikan program S1 dalam bidang Teknik Informatika, pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Shalawat beserta salam tidak pernah lupa kami haturkan pada junjungan kita, Nabi besar Muhammad SAW yang karenanya kita mendapat pencerahan menuju jalan yang lurus, jalan yang diridhoi dan bukan jalan orang sesat yang dimurkai. Semoga Allah melimpahkan atas mereka rahmat yang sesuai dengan keutamaan mereka sebagai pahala atas amal perbuatan mereka.

Dalam penyusunan skripsi ini, iringan do'a dan mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Prof. Dr. H. Mudjia Rahardjo, M.Si, selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Hj. Bayyinatul Muchtaromah, drh., M.Si, Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian, Ketua jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang

4. A'la Syauqi, M.Kom selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan, serta memberikan saran dan kepercayaan dalam dalam menyelesaikan tugas akhir ini.
5. Zainal Abidin, M.Kom selaku dosen pembimbing II yang telah memberi arahan, motivasi, masukan, saran serta bimbingan.
6. Dr. Suhartono, M.Kom selaku Dosen wali dan Dosen pembimbing yang memberikan semangat kepada penulis selama menjalani perkuliahan serta telah bersedia meluangkan waktu, tenaga, pikiran, arahan dan masukan yang sangat berguna dalam menyelesaikan skripsi ini.
7. Seluruh Dosen jurusan Teknik Informatika fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang yang telah mengalirkan ilmu, pengetahuan, pengalaman, wacana dan wawasannya, sebagai pedoman dan bekal bagi penulis.
8. Bapakku M. Shafwan (Alm.) dan Ibuku Umi Hidayati tersayang, kakak, adik-adikku dan seluruh keluarga besar di Takengon yang telah banyak memberikan doa, motivasi dan semangat, yang tiada henti sehingga penulisan skripsi ini dapat terselesaikan dengan baik. Perjuangan dan keikhlasan kalian membuat kami malu untuk tidak berprestasi dan berkarya.
9. Teman-teman seperjuangan *semantic* yaitu M. Edi Cahyono, M. Andy Nugroho yang senantiasa saling mendukung, membantu, menguatkan dan memberi semangat satu sama lain.
10. Teman-teman Jurusan Teknik Informatika fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang khususnya

angkatan 2007 yang telah memberi dukungan, motivasi, informasi, dan masukannya pada penulis selama menimba ilmu.

11. Segenap keluarga besar Ikatan Pemuda Pelajar Mahasiswa Tanah Gayo (IPPEMATANG) Malang yang telah memberikan nasehat, do'a serta pengalaman dan pembelajaran selama kuliah.

12. Kepada semua pihak yang telah membantu hingga terselesaikannya skripsi ini, baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari sepenuhnya bahwa sebagai manusia biasa tentunya tidak akan luput dari kekurangan dan keterbatasan. Maka dengan segenap kerendahan hati, penulis mengharapkan saran dan kritik yang dapat menyempurnakan penulisan ini sehingga dapat bermanfaat dan berguna untuk pengembangan ilmu pengetahuan. Teriring doa dan harapan semoga apa yang telah mereka berikan kepada penulis, mendapatkan balasan yang lebih baik dari Allah SWT. Amin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 23 November 2013

Penulis,

Agus Syodiqi

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TULISAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
ABSTRAK	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	7
1.6 Metodologi Penelitian	7
1.6.1 Pengumpulan Data.....	7
1.6.2 Perancangan Sistem.....	7
1.6.3 Pembuatan Aplikasi.....	8
1.6.4 Pengujian Aplikasi	8
1.6.5 Penyusunan Laporan	8
1.7 Sistematika Penulisan	9
BAB II TINJAUAN PUSTAKA.....	11
2.1 Perguruan Tinggi	12
2.1.1 Perguruan Tinggi Negeri (PTN)	12
2.1.2 Perguruan Tinggi Swasta (PTS).....	14
2.1.3 Program Studi (Prodi)	16
2.2 Semantik Web	17
2.2.1 Pengertian Semantik Web	17
2.2.2 Sejarah Semantik Web	19
2.2.3 Komponen Semantik Web	20
2.3 Penerapan Teknologi semantic web pada aplikasi perguruan tinggi	23
2.4 Ontology	25
2.4.1 Pengertian <i>Ontology</i>	25
2.4.2 Model <i>Ontology</i>	26
2.4.3 Tipe <i>Ontology</i>	27

2.4.4 Bahasa <i>Ontology</i>	27
2.4.5 Kegunaan <i>Ontology</i>	29
2.4.6 OWL (<i>Ontology Web Language</i>)	30
2.4.7 Protege.....	31
2.5 RDF (<i>Resource Description Framework</i>)	32
2.5.1 Pengertian RDF	32
2.5.2 RDF- Gravity	34
2.5.3 SPARQL.....	36
2.5.4 Jena <i>Framework</i>	38
2.5.5 SMORE (<i>Semantic Markup Ontology and RDF Editor</i>).....	39
2.5.6 Sphider	40
2.6 MCDM (<i>Multi Criteria Decision Making</i>).....	41
2.6.1 <i>Multi Objective Decision Making</i> (MODM)	45
2.6.2 <i>Multiple Atribute Decision Making</i> (MADM).....	45
2.6.3 Sifat-Sifat MCDM dala Memilih Kriteria.....	47
2.6.4 <i>Weighted Product Model</i> (WPM)	47
2.6.5 Memeilih Perguruan Tinggi	49
2.7 Korelasi Rekomendasi Perguruan Tingi Dalam Perspektif Al-qur'an dan Hadist	51
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	56
3.1 Tahapan Proses Pembuatan dengan <i>semantic web</i>	56
3.2 Pemodelan Data <i>Semantic</i> dengan <i>Ontology</i>	57
3.3 Perancangan <i>Onology</i>	60
3.3.1 Penentuan Konsep <i>Ontology</i>	61
3.3.2 Penentuan Daftar Terminologi	63
3.3.3 Defenisi Kelas dan Struktur Hirarki	63
3.3.4 Mendefinisikan Property dari Kelas (<i>Slot</i>)	66
3.3.5 Membuat Instance-instance	69
3.3.6 Visualisasi Graph <i>Ontology Gravity</i>	70
3.4 Membuat Metadata Menggunakan <i>Smore</i>	71
3.5 Membuat pemodelan dengan <i>N-triple</i>	73
3.6 <i>Indexing, Crawling</i> data menggunakan <i>Sphider</i>	75
3.7 Perancangan Aplikasi	77
3.7.1 Deskripsi Umum Sistem.....	77
3.7.2 <i>Context Diagram</i>	80
3.7.3 <i>Data Flow Diagram</i> (DFD).....	81
3.7.4 <i>Entity Relationship Diagram</i> (ERD)	82
3.7.5 <i>Flowchart Proses Pencarian</i>	83
3.8 Tahap Implementasi Metode MCDM dengan WPM	85
3.9 Desain <i>Interface</i>	88
3.9.1 Desain <i>interface</i> halaman awal.....	88
3.9.2 Desain <i>interface</i> halaman utama.....	90
3.9.3 Desain <i>interface</i> halaman detail	90
3.9.4 Desain <i>interface</i> login admin.....	91
3.9.5 Desain <i>interface</i> edit Data	92
3.9.6 Desain <i>Interface</i> halaman list Data.....	92
3.9.7 Desain <i>interface input</i> Data	93

3.10 Perancangan Ujicoba sistem.....	94
BAB IV IMPLEMENTASI DAN PENGUJIAN	96
4.1 Implementasi Sistem.....	96
4.1.1 Kebutuhan Perangkat Keras (<i>Hardware</i>).....	96
4.1.2 Kebutuhan Perangkat Lunak (<i>Software</i>)	97
4.2 Pengujian Aplikasi	97
4.2.1 Tampilan awal	98
4.2.2 Tampilan halaman Utama.....	99
4.2.3 Tampilan halaman detail	101
4.2.4 Interface Halaman Administrator	102
a. Login.....	102
b. Interface Halaman Panel Administrator	103
c. Halaman Admin.....	104
d. List Data Peguruan Tinggi.....	105
e. <i>Update/Add data/ dan Delete</i>	106
4.3 Implementasi Metadata	107
4.3.1 Hasil Proses <i>Crawling</i>	108
4.3.2 Proses <i>Indexing</i>	108
4.3.3 Pembuatan <i>table N-triple</i>	110
4.3.4 <i>Filter</i> kata utama.....	111
4.3.5 Pengindeksan data	112
4.3.6 Pencocokan Kata kriteria.....	113
4.4 Implementasi Antar Muka Sistem dengan MCDM	114
4.4.1 Menentukan Alternatif Keputusan.....	114
4.4.2 Penentuan Nilai Bobot.....	115
4.5 Uji Coba Pencarian	116
4.6 Evaluasi Sistem.....	120
BAB V PENUTUP	
5.1 Kesimpulan	121
5.2 Saran	121
DAFTAR PUSTAKA	122

DAFTAR TABEL

Tabel 2.1	Jumlah Perguruan Tinggi berdasarkan kabupaten di Jawa Timur	13
Tabel 2.2	Perbedaan MODM dan MADM	46
Tabel 3.1	Rancangan <i>Class</i> yang terbentuk pada <i>Class</i> Akreditasi	66
Tabel 3.2	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Alamat	66
Tabel 3.3	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Kabuten	66
Tabel 3.4	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Tahun	66
Tabel 3.5	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Jenjang_Pendidikan	67
Tabel 3.6	Rancangan <i>Class</i> yang terbentuk pada <i>subclass</i> jabatan_akademik <i>class</i> Pimpinan_perguruan_tinggi	67
Tabel 3.7	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Status	67
Tabel 3.8	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Kota	67
Tabel 3.9	Rancangan <i>Class</i> yang terbentuk pada <i>class</i> Periode	67
Tabel 3.10	Rancangan <i>Class</i> yang terbentuk pada <i>subclass</i> Lembaga_Pendidikan <i>Class</i> PerguruanTinggi	68
Tabel 3.11	Alternatif pencocokan Kriteria	86
Tabel 3.12	Rating kecocokan dari setiap alternatif	86
Tabel 4.1	Hasil Uji coba fungsional Sistem	117

DAFTAR GAMBAR

Gambar 2.1. Proses Aplikasi Web 2.0	20
Gambar 2.2. Proses Aplikasi Web 3.0	20
Gambar 2.3. Layer <i>Semantic Web</i>	21
Gambar 2.4. Hasil Pencarian Materi Perkuliahan	25
Gambar 2.5. Kegunaan <i>Ontology</i>	29
Gambar 2.6. Tampilan Protégé 3.4.5	32
Gambar 2.7. <i>Triple Graf RDF</i>	33
Gambar 2.8. <i>Screenshot RDF Gravity</i>	36
Gambar 2.9. <i>Query RDF menggunakan SPARQL</i>	38
Gambar 2.10. Tampilan <i>SMORE</i> versi 2.6.4.....	40
Gambar 2.11. Tampilan <i>Sphider</i>	41
Gambar 2.12. Matrik Keputusan MCDM	43
Gambar 2.13. <i>Multi Criteria Decision Making (MCDM) Tree</i>	44
Gambar 3.1. Tahapan proses dalam membangun <i>semantic web</i>	57
Gambar 3.2. Hirarki <i>is-a</i>	58
Gambar 3.3 Tahapan pemodelan <i>Ontology</i>	61
Gambar 3.4. Rancangan Struktur <i>Hirarki</i> Kelas Utama	64
Gambar 3.5 <i>Superclass</i> dan <i>subclass</i>	65
Gambar 3.6 Object Properties	69
Gambar 3.7 <i>Instance</i> Akreditasi_A dari Universitas	70
Gambar 3.8. <i>Skematisasi RDF-Gravity</i>	71
Gambar 3.9. <i>Alur</i> yang dilakukan dalam <i>SMORE</i>	71
Gambar 3.10. <i>individual</i> dalam <i>smore</i>	72
Gambar 3.11. Proses penerapan data RDF menggunakan <i>exhibit framewor</i>	73
Gambar 3.12. Konversi data menjadi <i>N-Triple</i>	74
Gambar 3.13. <i>Crawling</i> data pada <i>Sphider</i>	76
Gambar 3.14 ilustrasi proses <i>indexing</i>	76
Gambar 3.15. <i>Proses indexing</i> pada sebuah dokumen.....	77
Gambar 3.16. Skema Pendekatan <i>Semantic web</i>	79
Gambar 3.17. <i>Context diagram</i> aplikasi	80
Gambar 3.18. <i>Data Flow</i> diagram	81
Gambar 3.19. <i>Entity relationhsip diagram</i> menggunakan <i>Top braid composer</i>	82
Gambar 3.20 <i>Entity relationhsip diagram</i> menggunakan <i>RDF Gravity</i>	82
Gambar 3.20. Alur pencarian kriteria	84
Gambar 3.22. Desain <i>Interface</i> Halaman Pertama.....	88
Gambar 3.23. Desain <i>interface</i> halaman utama	90
Gambar 3.24. Desain <i>interface</i> halaman Detail	91
Gambar 3.25. Desain <i>Interface</i> Halaman Login Admin	91

ABSTRAK

Syodiqi, Agus. 2013. 07650003. Aplikasi Rekomendasi Perguruan Tinggi Berbasis Semantic Web Dengan Metode Multi Criteria Decision Making (MCDM).
Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
Pembimbing:(I) A'la Syaqui, M.Kom. (II) Zainal Abidin, M.Kom.

Kata Kunci: Perguruan Tinggi, *Semantic Web Ontology*, MCDM, N-Triple, OWL

Daya saing perguruan tinggi adalah kemampuan dari perguruan tinggi untuk menunjukkan keunggulan bersaing dan menawarkan nilai yang lebih atas kinerjanya dalam hal tertentu, dengan cara memperlihatkan situasi dan kondisi yang paling menguntungkan, dibandingkan dengan perguruan tinggi lainnya.

Dalam penelitian ini pencarian dilakukan dengan metode semantic web yang dimulai dengan pemodelan data semantic dalam bentuk ontology, perancangan ontology menggunakan Protégé 3.4.5. Hasil perancangan dan pengujian aplikasi *Decision Making* akan menghasilkan keputusan yang tepat dan fleksibel dengan aspek yang mempengaruhi keputusan tersebut, serta menghasilkan laporan agar *user* mengerti akan keputusan tersebut diambil.

Pemodelan dalam penggunaan semantic web ini, peneliti menggunakan pemodelan ontology. RDF (Resource Description Framework) serta OWL (Ontology Web Language) merupakan beberapa dukungan standart pada semantic web. Dengan improvisasi dalam proses pengaksesan data semantik, maka N-Triple yang digunakan tidak perlu menggunakan SPARQL sebagai query-nya, namun akan dikonversi dalam bentuk lebih sederhana berupa SPO (Subjek Predikat Objek) pada sebuah database dalam SQL sehingga dalam proses pengaksesan data akan menjadi lebih dinamis dan mudah.

Berdasarkan pengujian yang dilakukan dapat disimpulkan bahwa aplikasi pencarian menggunakan teknologi semantic web mampu melakukan pencarian sesuai konteks dari kalimat yang dimasukkan sebesar 84 %. Aplikasi rekomendasi perguruan tinggi ini berbasis pencarian menggunakan *semantic web* sebagai hasil dari uji coba dapat diperoleh prosentase kebenaran dari aplikasi ini yaitu 84%. Sehingga hasil pencarian yang didapatkan tepat dan akurat serta dapat menjadi lebih spesifik sesuai dengan penulisan kalimat yang dimasukkan untuk menentukan perguruan tinggi sesuai dengan kriteria-kriteria yang diinginkan.

ABSTRACT

Syodiqi, Agus. 2013. 07650003. *Application of Universities Recommendation Based Semantic Web Using Multi Criteria Decision Making (MCDM) Methods.* Informatics Engineering Department, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang.
Supervisor: (1) A'la Syauqi, M.Kom. (2) Zainal Abidin, M. Kom.

Keywords: University, *Semantic Web Ontology*, MCDM, N-Triple, OWL

University competitiveness is the ability of Universities to demonstrate its excellence in competitive and offer more value for its performance in a certain respect, by showing the situation and condition which the most favorable, compared to other universities.

In this research, a search is conducted using semantic web method that begins with semantic data modeling in the form of ontology, designing ontology using Protege 3.4.5. The results of the design and testing application of "Decision Making" will result in the right decision and flexible with the aspects that influence the decision, as well as generate reports in order that the users understand that the decision is taken.

Modeling in the use of the semantic web, the researcher is using ontology modeling. RDF (Resource Description Framework) and OWL (Web Ontology Language) are some standard supports on the semantic web. By improvisation in the process of data access semantics, then the N-Triple which is used don't need to use SPARQL as its query but it will be converted in a more simple form of SPO (Subject Predicate Object) on a database in SQL so that the process of accessing Data will be more dynamic and easier.

Based on the tests that have been performed can be concluded that the search application using semantic web technology is able to perform a search in context of the sentence which is entered in the amount of 84%. The application of Universities recommendation based searching using semantic web as a result of the test can be obtained percentage of correctness from this application is 84%. So the search results which are obtained are precise and accurate as well as can be more specific in accordance with the writing of sentences are included to determine the university in accordance with the desired criteria.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah menjadi komponen penting dalam berbagai bidang (Wellem, 2009) dalam perkembangannya teknologi informasi saat ini sudah tidak dapat lagi terlepas dari teknologi Internet. Internet telah menjadi suatu teknologi yang tidak dapat lepas dari kehidupan manusia modern sekarang ini. Salah satu teknologi yang saat ini dirasakan manfaatnya adalah *World Wide Web* (WWW) atau dikenal dengan nama *web*. *Web* kini menjadi sumber data yang terbesar dan sangat berharga untuk setiap pengguna karena didalam web kumpulan dokumen-dokumen saling terhubung dapat diakses melalui koneksi internet, dibutuhkan suatu teknologi untuk mendapatkan informasi yang benar-benar diinginkan tersebut.

Salah satunya dihadapi oleh siswa SMA adalah pengambilan keputusan ke perguruan tinggi, dimana seorang siswa yang telah lulus dari sekolah menengah atas dihadapkan pada pilihan-pilihan sulit. Seorang siswa dalam kehidupannya akan dihadapkan dengan sejumlah alternatif, baik yang berhubungan dengan kehidupan pribadi, sosial, belajar maupun karirnya. Adakalanya siswa mengalami kesulitan untuk mengambil keputusan dalam menentukan alternatif mana yang sebaiknya dipilih. Apakah nantinya akan meneruskan studi lanjut yakni melanjutkan pendidikan ke perguruan tinggi, atau akan bekerja maupun mengikuti pelatihan-pelatihan/ kursus.

Banyak siswa yang mengalami kekurangan informasi tentang studi lanjut terutama yang berkaitan dengan perguruan tinggi sehingga cenderung menjadi salah satu hambatan dalam menentukan arah pilih studi lanjut ataupun karirnya (Setyowati.dkk.2009). Siswa hanya memiliki informasi tentang perguruan tinggi negeri yang diminati, sedangkan kurang pengetahuan tentang perguruan tinggi swasta. Siswa masih bingung menentukan alternatif pilihan perguruan tinggi swasta dikarenakan banyaknya jenis-jenis perguruan tinggi swasta tersebut. Beberapa faktor yang dapat dijadikan dasar pertimbangan dalam mengambil keputusan memilih perguruan tinggi swasta adalah status perguruan tinggi, citra PTS, fasilitas fisik, biaya SPP, proses belajar mengajar, mutu dosen, mutu lulusan.

Maraknya perguruan tinggi berpotensi merosotnya mutu lulusan, mengingat standarisasi mutu lulusan tidak menjadi tujuan tetapi hanya dilihat dari aspek kuantitas yakni bagaimana mendapatkan jumlah mahasiswa sebanyak-banyaknya. Begitu juga dengan diberlakukannya otonomi kampus dimana perguruan tinggi negeri (PTN) dan swasta (PTS) memiliki kesamaan di dalam pengelolaan, sehingga ada kecenderungan untuk mencari dana yang memadai namun terkadang mengabaikan aspek mutu itu sendiri (Asmawi.2005).

Agar terhindarkan dari permasalahan tersebut maka para siswa perlu dibekali dengan informasi yang cukup dan akurat (Rindiani,dkk.2010). Sistem informasi ini bertujuan memberikan informasi secara lengkap tentang studi lanjut, dengan harapan agar siswa dapat merencanakan / mengambil keputusan secara tepat dan mantap sesuai dengan potensi yang dimiliki. Saat

ini Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) semakin berkembang pesat jumlahnya begitu juga dengan Prodi yang ada. Oleh karena itu, proses pengambilan keputusan perlu dimengerti oleh lulusan SMA sewaktu akan memasuki perguruan tinggi, karena melalui tahap-tahap tersebut keputusan yang akan diambilnya akan lebih efektif (Mahmudi.2006). Sebagai peluang potensial bagi dunia akademik dalam menyimpan koleksi dan menampilkan informasi Perguruan Tinggi atau Universitas seperti nama Universitas, Status, Akreditasi, Jurusan dan alamat perguruan tinggi.

Fitur-Fitur *Semantic Web* selalu dikembangkan para *developer* dalam rangka memberi kemudahan untuk *user* (pengguna) dari sistem. Sehingga, aplikasi rekomendasi pencarian mampu memproses informasi yang dapat mengarahkan *user* untuk memahami maksud dan tujuan yang diharapkan. Hal ini islam sangat menyarankan dan memperhatikan aktivitas memberi manfaat atau kemudahan untuk orang lain merupakan perbuatan sangat mulia. Sebagaimana nabi Muhammad SAW bersabda yang ditulis Syaikh Nashiruddin Al Albani dalam Ringkasan Kitab Shahih At-Tarhib Wa At-tarhib (<http://muslim.or.id>):

عن جابر قال : قال رسول الله صلى الله عليه وسلم : « المؤمن يألف ويؤلف ، ولا خير
« فيمن لا يألف ، ولا يؤلف ، وخير الناس أنفعهم للناس

Artinya: “*Jabir radhiyallau ‘anhuma berkata bahwa Rasulullah shallallahu ‘alaihi wasallam bersabda: “Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia.”* (HR. Thabrani dan Daruqutni)

Tentu saja manfaat dalam hadits diatas sangat luas. Dari hadits tersebut dijelaskan bahwa memberi manfaat atau kemudahan untuk orang lain adalah merupakan perbuatan yang sangat mulia. Sehingga Rosulullah menguatkannya dengan kalimat “خير” yang artinya sebaik-baik atau yang terbaik. Manfaat yang dimaksud bukan sekedar manfaat materi, namun manfaat yang bisa diberikan kepada orang lain bisa berupa ilmu umum yang diajarkan kepada orang lain dengan ilmu tersebut orang lain mendapatkan kemudahan.

Kemajuan *sains* dan teknologi telah memberikan kemudahan-kemudahan dan kesejahteraan bagi kehidupan manusia sekaligus merupakan sarana bagi kesempurnaan manusia sebagai hamba Allah dan khalifah-Nya karena Allah telah mengaruniakan anugerah kenikmatan kepada manusia yang bersifat saling melengkapi dan memberikan informasi (ilmu pengetahuan) yang bermanfaat. Sebagaimana firman Allah yang diturunkan kepada nabi Muhammad SAW:

قُلْ أَنْظُرُوا مَاذَا فِي السَّمَوَاتِ وَالْأَرْضِ وَمَا تُغْنِي الْآيَاتُ وَالنُّذُرُ عَنْ قَوْمٍ لَا

يُؤْمِنُونَ ﴿١٠١﴾

Artinya: *Katakanlah, Perhatikanlah apa yang ada di langit dan di bumi. Tidaklah bermanfaat tanda kekuasaan Allah dan rasul-rasul yang memberi peringatan bagi orang-orang yang tidak beriman.* (QS.Yunus : 101)

Dari ayat diatas para ulama' ahli tafsir menafsirkan tanda-tanda kebesaran Allah yang ada di alam ini bermanfaat bagi manusia. Oleh karena itu, umat manusia hendaknya mengambil manfaat dari tanda-tanda kekuasaan Allah swt. Allah memerintahkan kepada manusia agar melakukan pengkajian dan penelitian

terhadap alam semesta beserta segala isinya memerlukan kesungguhan manusia untuk meneliti atau melakukan eksperimen untuk dapat menyingkap isi kandungannya. Hal tersebut juga yang melatar belakangi manfaat yang dapat dirasakan dari hasil penelitian di Kopertis (Koordinasi Perguruan Tinggi Swasta) Jatim dibangunnya *serach engine* (mesin pencarian) untuk mengakses informasi dapat diperoleh dengan mudah.

Solusi yang dapat ditemukan dari permasalahan diatas adalah bagaimana menyimpan informasi tersebut secara lebih semantis. *Semantic web* dikatakan sebagai sebuah cara yang efisien untuk merepresentasikan data di *World Wide Web* sebagai sebuah database yang terhubung secara global (Ferdidkk:2007). Dengan adanya dukungan standard XML (*Extensible Markup Language*), RDF (*Resource Description Framework*), dan OWL (*Ontology Web Language*), semantic web mampu merepresentasikan data di web dalam suatu basis pengetahuan yang dapat diproses oleh mesin (komputer). Pendekatan *ontology* merupakan salah satu dasar struktur informasi yang dapat digunakan untuk melakukan proses pencarian lebih spesifik sesuai dengan keinginan *user* (Sabyala, 2012).

Proses pengambilan keputusan yang dilakukan *user* dengan mempertimbangkan banyak kriteria dengan tujuan untuk memberikan memudahkan dalam memilih perguruan tinggi. Pengambilan keputusan (*decision making*) adalah studi pengidentifikasian dan pemilihan alternatif-alternatif berdasarkan nilai-nilai dan preferensi pengambil keputusan. Proses pengambilan keputusan yang melibatkan banyak kriteria merupakan ranah kajian *Multi Criteria*

Decision Making (MCDM) dan bagaimana kerangka kerja (*framework*) pemilihannya (Andayani, dkk. 2012).

1.2 Rumusan Masalah

Dari latar belakang diatas maka rumusan masalah dalam penelitian ini adalah sejauh mana tingkat keberhasilan teknologi *semantic web* yang diterapkan pada sistem informasi aplikasi rekomendasi pencarian perguruan tinggi berbasis *semantic web* dengan dengan metode *Multi Criteria Decision Making* (MCDM)?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini antara lain:

- a. Menggunakan data objek perguruan tinggi, baik Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) Provinsi Jawa Timur
- b. Informasi mencakup perguruan tinggi diperoleh dari wilayah III PTN dan Kopertis wilayah VII Provinsi Jawa Timur
- c. Sistem difokuskan pada penerapan *semantic web* fasilitas pencarian perguruan tinggi dengan multi kriteria.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membangun Aplikasi rekomendasi perguruan tinggi menggunakan teknologi *semantic web* dengan *Multi Criteria Decision Making* (MDCM) dapat membantu menyelesaikan permasalahan sehingga jauh lebih mudah dan efisien.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah membantu para pengguna Sistem informasi Perguruan Tinggi serta Prodi Se-Jawa Timur dalam

memperoleh informasi berdasarkan kriteria yang sudah ditentukan sehingga akan mendapatkan hasil yang akurat yang relevan dan sesuai dengan keinginan.

1.6 Metode Penelitian

Dalam melakukan penelitian ini, peneliti membagi menjadi beberapa tahap pengerjaan yang digunakan sebagai acuan dalam penyelesaian penelitian tersebut. Berikut beberapa tahapan dalam melakukan penelitian:

1.6.1 Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Literatur

Pengumpulan data bertujuan untuk mempelajari teori-teori dengan membaca beberapa buku dan literatur yang berhubungan dengan permasalahan yang diambil. Khususnya kajian mengenai *Semantic Web* dan metode *Multi Criteria Decision Making* (MCDM).

b. Observasi

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

1.6.2 Perancangan dan Desain Sistem

Setelah mencari literatur dan memahami konsep-konsep dasarnya, dilakukanlah perancangan sistem terhadap konsep dasar tersebut. Pada proses ini menjelaskan bagaimana aplikasi sistem rekomendasi perguruan tinggi yang akan dibangun. Pada tahap perancangan sistem meliputi:

- a. Perancangan aplikasi pencarian berbasis teknologi *semantic web* ini menggunakan pendekatan ontologi, dimana di dalam *ontology* tersebut digunakan sebagai penyimpanan dokumen.
- b. Perancangan aplikasi terdiri atas perancangan proses-proses utama dan desain aplikasi yang terdiri atas desain menu pencarian rekomendasi perguruan tinggi dan desain utama dari aplikasi itu sendiri.
- c. Perancangan desain *user interface / form* yang digunakan pada aplikasi untuk menentukan kriteria pencarian perguruan tinggi

1.6.3 Pembuatan Aplikasi

Selanjutnya untuk menemukan informasi yang terdapat didalam ontologi dapat digunakan SPARQL sebagai bahasa *query* RDF/OWL, berfungsi sebagai *framework* yang menghubungkan antara ontologi dengan bahasa java menggunakan *tools* pendukung.

1.6.4 Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk memastikan agar sistem yang telah di buat sudah sesuai serta untuk mengetahui keberhasilan pembuatan sistem, apakah memenuhi spesifikasi dan akurasi yang direncanakan. Pada tahap ini dilakukan uji coba terhadap hasil implementasi yang didapat dari sistem dapat diimplementasikan untuk melakukan pencarian informasi perguruan tinggi.

1.6.5 Penyusunan Laporan

Setelah semua proses dilakukan, hal terakhir yang dilakukan penulis adalah menyusun laporan dari kegiatan penelitian. Penyusunan laporan

bertujuan untuk mendokumentasikan semua hasil penelitian yang dilakukan penulis.

1.7 Sistmatika Penulisan

Sistematika penulisan skripsi ini dengan melakukan pembagian bab sebagai berikut :

BAB I PENDAHULUAN

Pada bab pertama ini merupakan bab pembuka yang membahas dan menjelaskan tentang latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian manfaat penelitian, tahap-tahap penelitian, metode penelitian serta sistematika penyusunan laporan tugas akhir.

BAB II TINJAUAN PUSTAKA

Pada bab kajian pustaka berisi teori-teori yang menunjang penelitian ini, diantara adalah teori tentang perguruan tinggi negeri maupun swasta, *semantic web* dan metode yang digunakan *Multi Criteria Decision Making (MCDM)*.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan tentang proses-proses perancangan sistem dan analisa kebutuhan sistem untuk membuat sistem informasi perguruan tinggi Jawa Timur berbasis *semantic web* meliputi spesifikasi kebutuhan *software* dan langkah-langkah pembuatan aplikasi.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Menjelaskan tentang implementasi teknologi aplikasi rekomendasi perguruan tinggi berbasis *semantic web* untuk mesin pencarian dan pengujian penggunaan aplikasi meliputi konstruksi (*coding*), pengujian dan penggunaan sistem serta meliputi cara kerja program.

BAB V : KESIMPULAN DAN SARAN

Merupakan kesimpulan dari pembahasan pada bab-bab sebelumnya, dan saran-saran yang ditujukan kepada pihak yang terkait sehubungan dengan hasil sistem.

LAMPIRAN

Lampiran merupakan data pelengkap dari hasil olahan yang menunjang penulisan *tugas akhir* berfungsi untuk melengkapi uraian yang telah disajikan dalam bagian utama ditempatkan di bagian ini.

BAB II

TINJAUAN PUSTAKA

2.1 Perguruan Tinggi

Perguruan tinggi adalah satuan pendidikan penyelenggara pendidikan tinggi. Peserta didik perguruan tinggi disebut mahasiswa, sedangkan tenaga pendidik perguruan tinggi disebut dosen. Keberadaan perguruan tinggi di Indonesia selama ini memiliki dualism (Fadjar,2007) yaitu, Perguruan Tinggi Negeri (PTN) dan Perguruan Tinggi Swasta (PTS). Perguruan Tinggi sebenarnya tidak lepas dari konteks implementasi desentralisasi penyelenggaraan pendidikan, sebagai akibat dari berlakunya UU No. 33/2004 tentang Perimbangan Keuangan antara Pemerintah Pusat dan Daerah. Menurut Irianto (2011) Implikasi pada sistem manajemen pendidikan tinggi menuntut apa yang disebut dengan otonomi Pendidikan Tingkat Tinggi yang diterjemahkan dengan salah satu model konsep Badan Hukum Milik Negara (BHMN).

Hal ini masih ditambah lagi dengan kondisi dimana persaingan perguruan tinggi semakin ketat. Sudah barang tentu universitas yang mampu menanamkan citra positifnya pada khalayaklah yang akan memperoleh banyak perhatian, sehingga menjadi tujuan calon mahasiswa untuk melanjutkan pendidikannya. Namun citra positif tidak dapat secara otomatis terwujud bersamaan dengan terciptanya kualitas dan pengelolaan yang baik (Ike.2007).

Pendidikan mempunyai peranan yang sangat sentral dan strategis, terutama jika dikaitkan dengan upaya peningkatan mutu sumber daya manusia (SDM). Karena hanya dengan sumber daya manusia yang berkualitaslah akan tercipta peningkatan harkat dan martabat manusia yang sejati. Hakekat dari pendidikan sebagai sebuah usaha “*the acquisition of knowledge*” merupakan suatu perjalanan hidup yang tiada henti, dimulai semenjak seseorang dilahirkan hingga yang bersangkutan kembali ke pangkuan Sang Pencipta (Indrajit.2002).

Perguruan Tinggi sebagai salah satu bagian penting dalam dunia pendidikan yang ikut bertanggungjawab dalam upaya mencerdaskan kehidupan bangsa mempunyai tanggungjawab dan peran yang sangat strategis untuk mengambil bagian dalam mengatasi permasalahan kualitas sumber daya manusia. Kebijakan pemerintah yang memberikan kesempatan seluas-luasnya kepada semua komponen masyarakat untuk berpartisipasi dalam pembangunan pendidikan di Indonesia serta untuk memberdayakan peran serta masyarakat menyelenggarakan pendidikan berdasarkan prinsip otonomi dalam konteks negara kesatuan Republik Indonesia khususnya pendidikan tinggi (Sawaji, dkk.2010).

2.1.1 Perguruan Tinggi Negeri (PTN)

Perguruan tinggi Negeri dalam kedudukannya pemerintah, dituntut untuk menjalankan tata-pamong yang baik (*good governance*) seperti keharusan menunjukkan transparansi, akuntabilitas publik, dan partisipatori (Raharjo, 2010). Perguruan tinggi di Indonesia mengenal bebarapa istilah yang digunakan untuk menyebut perguruan tinggi, secara umum istilah tersebut tidak memiliki

perbedaan satu sama lain. Ada beberapa istilah yang sering disebut yaitu; Universitas, Insitut, Sekolah Tinggi, Akademi dan Politeknik.

Daftar nama – nama perguruan tinggi dengan status perguruan tinggi yang berada di wilayah propinsi Jawa Timur, baik berupa instiut, politeknik, akademik, sekolah tinggi dan universitas.

Table 2.1 Jumlah Perguruan Tinggi berdasarkan Kabupaten di Jawa Timur

Sumber data dari: (<http://www.kopertis7.go.id>)

NO	KOTA	BENTUK PERGURUAN TINGGI								JUMLAH					
		UNIV		INS T	SEKTI		AKA D	POLTE K		P.STUDI			PTS		
		PT	P T	PS	P T	PS	PT	PS	PT	PS	PT	P S	%	P T	%
1	Bangkalan					9	3	2	2			11	0.7 1%	5	1.52 %
2	Banyuwangi	37	3			5	2	3	2	3	1	48	3.0 8%	8	2.44 %
3	Batu							3	1			3	0.1 9%	1	0.3 %
4	Blitar	17	1			10	4	2	1			29	1.8 6%	6	1.83 %
5	Bojonegoro	5	1	6	1	4	2	2	1			17	1.0 9%	5	1.52 %
6	Bondowoso	7	1			0		1	1			8	0.5 1%	2	0.61 %
7	Gresik	20	2			0	6	2	4	4		30	1.9 2%	8	2.44 %
8	Jember	41	3	8	1	12	6	8	7			69	4.4 3%	17	5.18 %
9	Jombang	27	2			0	24	7	3	3		54	3.4 6%	12	3.66 %
10	Kediri	55	4	15	1	26	12	12	10	4	2	11 2	7.1 8%	29	8.84 %
11	Lamongan	24	2			0	8	4				32	2.0 5%	6	1.83 %
12	Lumajang	3	1			0	6	4				9	0.5 8%	5	1.52 %

13	Madiun	34	3	12	1	5	4	6	4	6	1	63	4.04%	13	3.96%
14	Magetan					4	1	1	1			5	0.32%	2	0.61%
15	Malang	190	11	33	4	80	28	8	8	8	3	319	20.46%	54	16.46%
16	Mojokerto	17	2			11	5	6	5	2	1	36	2.31%	13	3.96%
17	Nganjuk					11	4	1	1			12	0.77%	5	1.52%
18	Ngawi	6	1			5	1	1	1			12	0.77%	3	0.91%
19	Pacitan		0			4	1					4	0.26%	1	0.3%
20	Pamekasan	23	2									23	1.48%	2	0.61%
21	Pasuruan	15	2			15	5	2	2			32	2.05%	9	2.74%
22	Ponorogo	21	2			2	1	2	2			25	1.6%	5	1.52%
23	Probolinggo	10	1			9	5	5	3			24	1.54%	9	2.74%
24	Sampang					2	1	2	2			4	0.26%	3	0.91%
25	Sidoarjo	21	2			9	4	8	7			38	2.44%	13	3.96%
26	Situbondo	6	1			1	1	4	3			11	0.71%	5	1.52%
27	Sumenep	9	1			6	1					15	0.96%	2	0.61%
28	Surabaya	313	24	31	4	96	31	17	14	17	4	474	30.4%	77	23.48%
29	Trenggalek					2	1					2	0.13%	1	0.3%
30	Tuban	22	2			4	2					26	1.67%	4	1.22%
31	Tulungagung	6	1			6	2					12	0.77%	3	0.91%
Jumlah		929	75	105	12	382	144	103	85	40	12	1887			

2.1.2 Perguruan Tinggi Swasta (PTS)

Perguruan Tinggi Swasta (PTS) merupakan salah satu organisasi sosial yang bergerak dalam bidang edukasi menuju penciptaan sumber daya manusia yang berkompeten dalam segala dimensi kehidupan. Perguruan tinggi swasta harus menerapkan sudut pemikiran baru yang mengandung unsur fleksibilitas, kecepatan, inovasi, dan integrasi. Fleksibilitas, kecepatan, inovasi dan integrasi sangat memerlukan sumberdaya manusia yang penuh dengan kreativitas.

Perguruan tinggi swasta (PTS) memiliki beberapa kesamaan dengan perusahaan swasta karena dikelola dan dikembangkan dengan praktek-praktek yang sama dengan perusahaan swasta karena memang merupakan bentuk perguruan tinggi yang diharapkan dapat mandiri tanpa sokongan besar dari pemerintah. Namun juga memiliki perbedaan yaitu memiliki beberapa bagian yang bersifat non-profit, keuntungan yang diperoleh PTS pada akhirnya akan dicoba dikembalikan kepada masyarakat melalui peningkatan kualitas pendidikan, pemberian beasiswa, peningkatan fasilitas, pengabdian masyarakat, bahkan peningkatan keilmuan itu sendiri.(Sutjiati:2010).

PTS memiliki lembaga dibawah Kemdiknas yaitu, Kopertis mempunyai tugas melaksanakan kebijaksanaan pembinaan terhadap perguruan tinggi swasta secara operasional di wilayah kerjanya dengan mendapat bantuan teknis akademik dari perguruan tinggi negeri (<http://www.pts.co.id>).

Untuk menyelenggarakan tugas tersebut, Kopertis mempunyai tugas:

- a. Melaksanakan bimbingan penyelenggaraan program Tri Dharma Perguruan Tinggi pada perguruan tinggi swasta di wilayah kerjanya.
- b. Memberi dorongan dan saran-satan dalam rangka perguruan tinggi swasta sesuai dengan kebijaksanaan yang ditentukan Jenderal Perguruan Tinggi.
- c. Memberikan bantuan sarana dan tenaga kepada perguruan tinggi swasta dalam rangka peningkatan kemampuan untuk mandiri.
- d. Melaksanakan ujian negara bagi mahasiswa perguruan tinggi swasta di wilayah kerjanya.
- e. Melaksanakan pengendalian teknis dan pengayoman kepada perguruan tinggi swasta di wilayah kerjanya.
- f. Melaksanakan tugas lain atas petunjuk Direktur Jenderal Pendidikan Tinggi.

Wilayah kerja kopertis di Indonesia dibagi menjadi 12 wilayah yang mencakup 33 propinsi. Pada penelitian ini penulis mengambil data pada Kopertis Wilayah VII di Surabaya, wilayah kerja meliputi propinsi Jawa Timur.

2.1.3 Program Studi (Prodi)

Program studi merupakan penataan program akademik bagi bidang studi tertentu yang didedikasikan untuk (Kemdiknas, 2011): (1) menguasai, memanfaatkan, mendiseminasikan, mentransformasikan dan mengembangkan ilmu pengetahuan, teknologi, dan seni (Ipteks) dalam bidang studi tertentu, (2) mempelajari, mengklarifikasikan dan melestarikan budaya yang berkaitan dengan bidang studi tertentu, serta (3) meningkatkan mutu kehidupan masyarakat dalam kaitannya dengan bidang studi tertentu.

Oleh karena itu program studi sebagai lembaga melaksanakan fungsi Tridarma Perguruan Tinggi, yaitu pendidikan, penelitian dan pengabdian kepada masyarakat, serta mengelola *Ipteks* selaras dengan bidang studi yang dikelolanya. Untuk menopang dedikasi dan fungsi tersebut, program studi harus mampu mengatur diri sendiri dalam upaya meningkatkan dan menjamin mutu secara berkelanjutan, baik yang berkenaan dengan masukan, proses maupun keluaran program akademik dan layanan yang diberikan kepada masyarakat selaras dengan bidang studi yang dikelolanya.

Menentukan jurusan kuliah termasuk salah satu keputusan besar dalam kehidupan seseorang. Pasalnya, keputusan tersebut biasanya berpengaruh besar bagi perjalanan karier dan masa depan seseorang (<http://edukasi.kompas.com>). Berdasar data Koordinasi Perguruan Tinggi Swasta (Kopertis) Wilayah VII Jawa Timur (Jatim), setidaknya ada 367 Program Studi (Prodi) di berbagai PTS se-Jatim masa akreditasinya telah kedaluwarsa, di Jatim terdapat 1.569 Perguruan Tinggi Swasta 39% Prodi PTS Tak Terakreditasi (<http://www.surabayapost.co.id>).

2.2 *Semantic Web*

2.2.1 *Pengertian Semantic Web*

Secara garis besar web semantik adalah informasi dalam jumlah sangat besar di *World Wide Web*. *Semantic web* adalah perkembangan generasi berikutnya atau yang biasa disebut sebagai evolusi dari WWW (*World Wide Web*), yang dicetuskan pada tahun 2002. *Semantic web* didefinisikan sebagai

sekumpulan teknologi, dimana memungkinkan komputer memahami arti dari sebuah informasi berdasarkan *metadata*, yaitu informasi mengenai informasi.

Dengan adanya *metadata*, komputer diharapkan mampu mengartikan hasil pencarian menjadi lebih detail dan tepat. W3C (*World Wide Web Consortium*) mendefinisikan format *metadata* tersebut adalah *Resource Description Framework (RDF)*. Tiap unit dari RDF adalah 3 komposisi, yaitu *subject*, *predicate*, dan *object*. *Subject* dan *object* adalah entitas yang ditunjukkan oleh teks. Sedangkan *predicate* adalah komposisi yang menerangkan sudut pandang dari *subject* yang dijelaskan *object*. Hal yang paling menarik dari RDF yaitu *object* dapat menjadi *subject* yang nantinya diterangkan oleh *object* yang lainnya. Sehingga *object* atau masukan dapat diterangkan secara jelas dan detail, serta sesuai dengan keinginan pengguna yang memberikan masukan (Cahyono,2012).

Dengan metode web semantik data berbasis HTML dapat dirubah menjadi format yang dapat dipahami oleh mesin, sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan antara informasi. Web semantik mampu melakukan perubahan ini dengan bantuan XML (*Extensible Markup Language*) dan data language standard seperti RDF (*Resource Description Framework*) dan OWL (*Ontology Web Language*), dua standarisasi dari W3C (*World Wide Web Consortium*). Dengan berbagai standar tersebut, memungkinkan pengembang web (*Web Developer*) untuk menambahkan satu layer "arti" pada dokumen webnya.

Sebagai *framework* untuk mendefinisikan bagaimana beberapa data terhubung dan bagaimana relasi yang menyertai data-data tersebut seharusnya ditampilkan. Dalam mencapai tujuannya dibutuhkan pemberian *meaning* ke dalam masing-masing *content* (sebagai atribut) yang akan digunakan oleh teknologi *web semantic* ke dalam beberapa *layer* (Suteja, dkk. 2008).

2.2.2 Sejarah Semantic Web

Istilah *Semantic Web* pertama kali diperkenalkan oleh *Tim Berners-Lee Hender*, dan *Ora Lasilla*, di majalah *Scientific American* pada tahun 2001. Pada saat itulah *Tim Barners-Lee* sebagai direktur W3C mulai membuat infrastruktur *Semantic Web* yang terus dikembangkan sampai sekarang. *Semantic web* disebut dengan istilah web 3.0 merupakan perkembangan dari teknologi Web dari masa-masa sebelumnya. Para ahli telah melakukan klasifikasi terhadap perkembangan teknologi web yang digunakan pada umumnya. Mereka membagi teknologi yang digunakan dengan tiga kategori, antara lain:

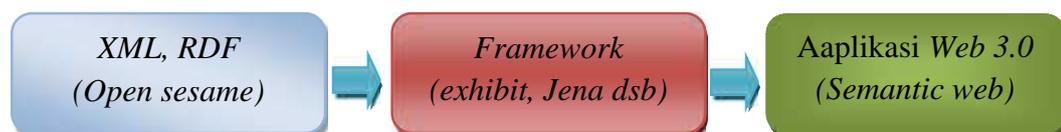
- a. Web 1.0 merupakan teknologi web generasi pertama yang merupakan revolusi baru di dunia internet karena telah mengubah cara kerja dunia industri dan media. Pada intinya Web 1.0 didesain untuk keperluan akses informasi dan memiliki sifat yang sedikit interaktif.
- b. Web 2.0 Inovasi dalam dunia web semakin hari kian mengalami perkembangan yang berarti, ini dibuktikan dengan adanya Teknologi Web 2.0 yang dikembangkan sekitar tahun 2004. Didesain untuk lebih berorientasi kepada kolaborasi dan sharing informasi secara online. Fitur-fitur yang

tersedia dalam teknologi Web 2.0 antara lain: CSS (*Cascading Style Sheet*), aplikasi Rich Internet atau berbasis AJAX, *markup* XHTML, sindikasi dan agregasi data menggunakan RSS, valid URL, XML *Web-Service* API, dan *Folksonomies*. Berikut ini gambar 2.1 adalah ilustrasi teknologi web 2.0:



Gambar 2.1 Proses Aplikasi Web 2.0

- c. Web 3.0 merupakan pengembangan teknologi selanjutnya dan mencoba menyempurnakan Web 2.0 dengan memberikan penekanan penelitian pada *Semantic Web*, *Ontology*, *Web Service*, dan *Social Software*. Perbedaan yang paling mendasar antara teknologi Web 3.0 dengan sebelumnya adalah pada kemampuan mesin dalam melakukan pengambilan keputusan dalam hal ini memahami makna dari sebuah informasi yang tidak dapat dilakukan oleh teknologi Web sebelumnya teknologi Web 3.0. Gambar 2.2 berikut adalah ilustrasi Web 3.0:

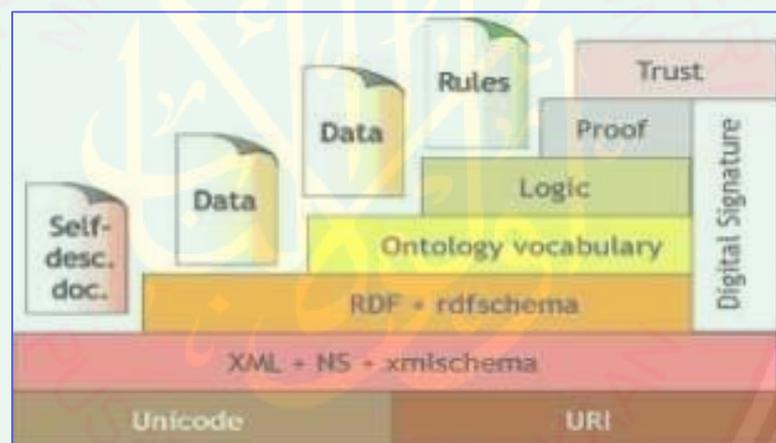


Gambar 2.2 Proses Aplikasi Web 3.0 (*semantic web*)

2.2.3 Komponen Semantic Web

Komponen-komponen utama dalam *semantic web* antara lain URI (*Uniform Resource Identifier Protocol*). Komponen tersebut dikenal dengan komponen yang disepakati oleh W3C sebagai sebuah standar untuk keperluan data agar bisa dipahami oleh mesin diantaranya adalah RDF (*Resource Description Framework*) serta OWL (*Ontology Web Language*).

Secara garis besar layer Semantic Web dapat dilihat pada gambar berikut pada gambar 2.3 berikut:



Gambar 2.3 Layer semantic web

(Sumber: Ibrahim, 2007)

Secara komprehensif penjelasan masing-masing komponen dan hubungan keterkaitannya dapat dilihat dibawah:

a. Unicode dan URI

Layer paling bawah adalah layer unicode dan URI (*Uniform Resource Identifiers*) yang memastikan penggunaan sekumpulan karakter yang telah

disepakati secara internasional dan menyediakan alat untuk mengidentifikasi obyek di semantic web

b. XML, NS dan XML(S)

XML (*Extensible Markup Language*) dan *Namespace* serta *Schema*, merupakan sintaks yang berfungsi untuk menyajikan struktur data pada *web*.

c. RDF+RDF(S)

RDF (*Resource Description Framework*) merupakan model berbentuk *graph* untuk merepresentasikan *resource* dan relasinya. Sedangkan RDF *Schema* adalah definisi kosakata yang digunakan pada RDF.

d. *Ontology Vocabulary*

Bahasa *ontology* yang direkomendasikan oleh W3C pada 10 Februari 2004 adalah OWL *Web Ontology Language*, merupakan bahasa yang lebih kaya dan kompleks untuk mendeskripsikan *resource*.

e. *Logic dan Proof*

Layer ini berupa *rule* dan *System* untuk melakukan *reasoning* pada *ontology* sehingga dapat disimpulkan apakah suatu *resource* memenuhi syarat tertentu.

f. *Trust*

Layer dari *semantic web* yang memungkinkan pengguna *web* untuk mempercayai suatu informasi pada *web* atau *layer trust* mengevaluasi apakah hasil dari *layer proof* bisa dipercaya (Sulhan, 2010).

g. *Digital Signatur*

Merupakan yang mendefinisikan blok dari data yang tereksripsi yang akan dimanfaatkan oleh komputer dan agen untuk memastikan apakah suatu informasi disediakan oleh sumber yang terpercaya serta mendeteksi adanya perubahan pada dokumen.

2.3 Penerapan Teknologi Semantic Web Pada Aplikasi Rekomendasi Perguruan Tinggi

Pengembangan aplikasi rekomendasi pencariin perguruan tinggi berbasis *semantic web* dengan metode banyak kriteria. Pada skripsi ini pada proses penyimpanan menggunakan MYSQL sampai database yang telah di *crawling* akan disimpan dengan menggunakan *tools frame work* berbasis java.

Semantic web dengan bentuk ontologinya dirancang dengan tujuan untuk digunakan memproses isi informasi informasi untuk memudahkan manusia. Dengan adanya Metode MCDM membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria, pihak yang berkepentingan, hasil dan dengan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas.

Dalam salah satu penelitian terkait tentang pencarian berbasis *semantic web* yang dikerjakan oleh Ferdila dkk. 2010 dengan judul “ Aplikasi Web Semantik Untuk Pencarian Materi Perkuliahan (Studi kasus sistem informasi di Universitas Gunadarma)”, digunakan aturan sebagai berikut:

- a. Aplikasi ini dibuat menggunakan teknologi web semantik. Dalam penulisan ini komponen-komponen teknologi web semantik yang dipergunakan antara lain RDF (*Resource Description Framework*) dipergunakan sebagai representasi pengetahuan yang digunakan.
- b. SPARQL yang dipergunakan sebagai *query* untuk mengambil informasi yang terdapat pada RDF.
- c. RAP (*RDF API for PHP*) yang dipergunakan untuk menjembatani antara RDF dengan PHP sehingga informasi yang ada di dalam RDF dapat digunakan pada PHP.

Proses pengerjaan aplikasi ada beberapa tahap. Bagian-bagian penting dalam model RDF yaitu *Resource* yang digunakan untuk menggambarkan apa saja yang dapat dimiliki oleh sebuah URI, *Property* berisikan nama *property-property* yang ada dalam *resource*, *property value* yang berisikan nilai dari sebuah *property*. Model data RDF yang telah terbentuk terdiri dari statemen yang di dalamnya terdiri dari subjek, predikat dan objek. *Query* SPARQL terdiri dari beberapa bagian utama yaitu bagian *select*, *from* dan *where*. Pada bagian *select* berisikan variabel yang digunakan untuk mengambil nilai yang diinginkan.

Kesimpulan dari yang di uji cobakan Aplikasi pencarian materi perkuliahan ini dapat melakukan pencarian materi perkuliahan dengan menggunakan kata kunci berupa judul materi, mata kuliah serta nama dosen yang mengajar materi tersebut. Pencarian juga dapat dilakukan dengan menggunakan satu kata kunci atau beberapa kata kunci. Berikut gambar 2.4 *interface* aplikasi

pencarian materi perkuliahan di jurusan teknologi informasi Universitas Gunadarma yang berhasil dibuat:



Gambar 2.4 Hasil Pencarian Materi Perkuliahan
(Sumber: ferdila, dkk. 2010)

2.4 *Ontology*

2.4.1 *Pengertian Ontology*

Ontology merupakan sebuah pendekatan yang sering dipakai dalam pengembangan sistem berbasis *knowledge management*. Paradigma *ontology* membuka kemungkinan untuk berpindah dari pandangan berorientasi dokumen ke arah pengetahuan yang saling terkait, dapat dikombinasikan, serta dapat dimanfaatkan kembali secara lebih fleksibel dan dinamis. Chandrasekaran dan Josehson (1999) mendefinisikan *ontology* sebagai cara merepresentasikan pengetahuan tentang makna objek, properti dari suatu objek, serta relasi objek tersebut yang mungkin terjadi pada domain pengetahuan.

Ontology sangat penting karena dapat digunakan untuk menerangkan tentang struktur suatu disiplin ilmu. *Ontology* membuka kemungkinan untuk berpindah dari pandangan berorientasi dokumen, ke arah pengetahuan yang saling terkait, dikombinasikan dan untuk digunakan di kemudian hari. Secara umum, *ontology* digunakan *Artificial Intelligence* (AI) dan presentasi pengetahuan. Segala bidang ilmu yang ada di dunia, dapat menggunakan metode *ontology* untuk dapat berhubungan dan saling berkomunikasi dalam hal pertukaran informasi antar sistem-sistem yang berbeda.

2.4.2 Model *Ontology*

Secara teknis sebuah *Ontology* dapat dimodelkan atau direpresentasikan dalam bentuk *classes*, *properties*, *slots* dan *instances*.

- a. *Classes*, menerangkan konsep atau makna suatu *domain*. *Class* merupakan kumpulan elemen dengan properti yang sama. Suatu *class* dapat mempunyai turunan *subclass* yang menerangkan konsep secara lebih spesifik.
- b. *Properties*, menerangkan konsep nilai-nilai, status, terukur yang mungkin ada untuk *domain*.
- c. *Slots*, merupakan representasi dari kerangka pengetahuan atau relasi yang menerangkan property dari *class* dan *instance*.
- d. *Instances*, merupakan individu yang telah dibuat. *Instances* dari sebuah *subclass* merupakan *instance* dari suatu *superclass*.

2.4.3 Tipe *Ontology*

Ontology dapat dibedakan menjadi beberapa tipe sebagai berikut:

a. *Upper-level ontology*

Upper-level Ontology merupakan *Ontology* yang berupa suatu model umum untuk merepresentasikan apa yang ada di dunia, sangat serupa dengan apa yang diteliti dalam filosofical theory. Saat ini ada SUO (*Standard Upper Ontology*) yang dikembangkan oleh IEEE. Namun sangat sulit untuk mencapai kesepakatan dalam menetapkan *Ontology* yang demikian umum.

b. *Domain ontology*

Domain ontology merupakan *Ontology* yang merepresentasikan suatu domain tertentu saja. Banyak penelitian yang mengembangkan *Ontology* di bidang kesehatan atau biologi, seperti *Gene Ontology*, *Cancer Ontology*, dan *Medical Ontology*.

c. *Application dan Task Ontology*

Application dan Task Ontology merupakan *Ontology* yang khusus menyatakan *application* dan *task* yang independen terhadap domain. Contoh *Ontology* tipe ini adalah PROTON yang digunakan untuk *knowledge management system* dan selanjutnya dikembangkan pula untuk *automatic entity recognition* dan *information extraction* dari teks.

2.4.4 Bahasa *Ontology*

Ontology menggunakan banyak variasi struktur, tergantung dari penggunaan bahasa ontology termasuk sintaksis yang digunakan. Perlu diingat adalah *ontology* tidak melakukan apapun, fungsi perhitungan dan lainnya yang memproses *ontology* tidak hanya tergantung dari data yang terdapat dalam

ontology tersebut, tetapi juga tergan-tung kepada aplikasi yang digunakan. Selain itu, unsur logika dalam *ontology* diinterpretasikan dalam bahasa representasi sehingga *semantic* bersifat formal dan dapat digunakan oleh mesin misalnya untuk inferensi atau penarikan kesimpulan baru (Sulhan, 2010).

Untuk dapat digunakan, sebuah *ontology* harus diekspresikan dalam notasi yang nyata. Sebuah bahasa *ontology* adalah sebuah bahasa formal dari sebuah pengembangan *ontology*. Beberapa komponen yang menjadi struktur *ontology*, antara lain:

- a. *XML (Extensible Markup Language)* Menyediakan sintaksis untuk *output* dokumen terstruktur, tetapi belum dipaksakan untuk dokumen XML menggunakan *semantic constrains*.
- b. *XML Schema* Bahasa untuk pembatasan struktur dari dokumen XML
- c. *RDF (Resource Description Framework)* Model data untuk objek ('resources') dan relasi diantaranya, menyediakan *semantic* yang sederhana untuk model data tersebut, dan data model ini dapat disajikan dalam *sintaksis XML*.
- d. *RDF Schema* Adalah kosakata untuk menjelaskan *properties* dan *classes* dari sumber RDF, dengan sebuah *semantics* untuk hirarki penyamarataan dari *properties* dan *classes*.
- e. *OWL (Ontology Web Language)* Menambahkan beberapa kosakata untuk menjelaskan *properties* dan *classes*, antarlain : relasi antara *classes* (misalkan *disjointness*), kardinalitas (misalkan 'tepat satu'), *equality*, berbagai

tipe dari *properties*, karakteristik dari *properties* (misalkan *symmetry*), menyebutkan satu persatu *classes*.

2.4.5 Kegunaan *Ontology*

Kegunaan *Ontology* secara umum antara lain sebagai *controlled vocabulary*, *semantic interoperability*, *knowledge sharing*, dan *resuse*. Seperti yang diperlihatkan pada gambar 2.5 berikut:



Gambar 2.5 Kegunaan *Ontology*

(Sumber: <http://qlitee.com/W3C>)

Dari gambar diatas penjelasan dari ketiga katagori kegunaan *ontology* sebagai berikut:

a. *Communication*

Model ini dibuat dengan menyusun defenisi, kosakata, dan terminologi yang secara informal merepresentasikan semantik. Bentuk visual seperti diagram *Entity Relationship* (ER) dan *Unified Modeling Language* (UML) yang efektif untuk memahami sistem.

b. *Interoperability*

Ontology juga sangat bermanfaat untuk integritas sistem sudah ada, terlebih lagi untuk aplikasi sistem terdistribusi. Keragaman informasi di

level sintaksis dan struktur dapat diatasi dengan *Ontology* level semantik sebagai format standar data.

c. *System Engineering*

Ontology berguna untuk membuat spesifikasi sistem terdiri atas komponen relasinya.

2.4.6 OWL (*Ontology Web Language*)

OWL adalah bahasa *ontology* yang baru untuk sebuah *semantic web*, dikembangkan oleh *World Wide Web Consortium* (W3C) kelompok kerja *Web Ontology*. Pada mulanya OWL didesain untuk merepresentasikan informasi tentang kategori dari sebuah objek dan bagaimana objek tersebut berhubungan. OWL dapat juga menyediakan informasi tentang objek itu sendiri. Sebagai hasil usaha yang dilakukan oleh kegiatan *Semantic Web* W3C, OWL harus cocok dengan visi *Semantic Web* yaitu bahasa.

OWL menambahkan kosakata untuk menggambarkan kelas dan *properti*, yaitu antara lain: relasi antar kelas misalnya *disjoint*, *kardinalitas*, *equality*, tipe *properti* yang lebih kaya, karakteristik *properti* misalnya simetri, dan kelas enumerasi. Dengan demikian OWL memiliki fasilitas lebih dibanding XML dan RDF/S dalam mengekspresikan arti dan semantik. *Ontology* OWL merupakan graf RDF yang diubah menjadi kumpulan *triple* RDF.

2.4.7 Protégé

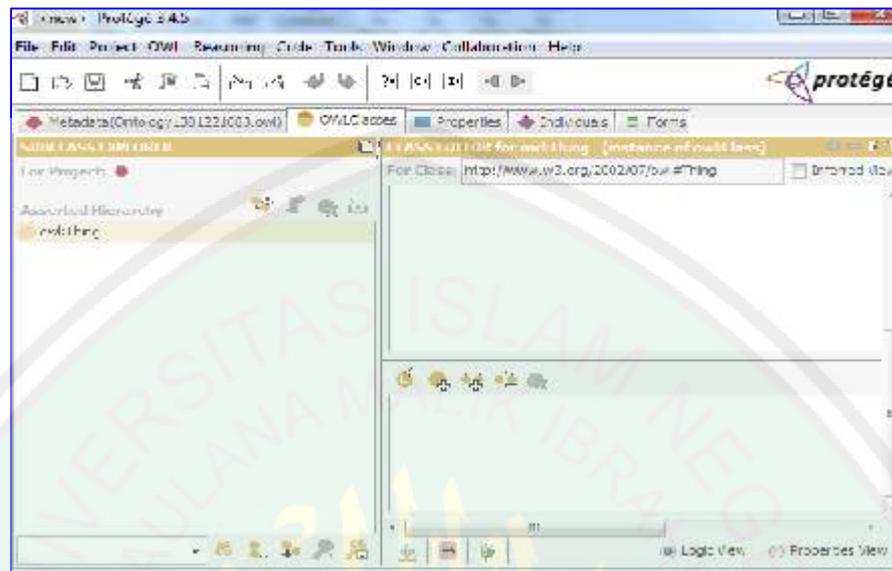
Untuk mengimplementasikan konsep *Ontology* dari pemodelan sistem rekomendasi perguruan tinggi, maka Penulis menggunakan perangkat lunak

Protégé 3.4.5 sebagai alatbantu. Installer untuk perangkat lunak tersebut dapat di-*download* pada *web site* <http://protege.stanford.edu>. Perangkat lunak ini dapat berjalan dengan baik pada *platform* Windows, Mac OS, Solaris, Linux Unix. Proses instalasi protégé ini cukup mudah karena telah disediakan wizard yang akan memandu setiap langkah para pemakai. Paket installer protégé ini juga telah menyediakan Java *Virtual Machine* (VM) yang merupakan *interpreter* dari aplikasi protégé.

Protégé adalah perangkat lunak yang digunakan untuk pengembangan *Knowledge Base System*. Protégé merupakan sebuah alat bantu yang digunakan untuk membuat sebuah *domain ontology*, menyesuaikan *form* untuk *entry* data, dan memasukkan data. Berbagai format penyimpanannya adalah OWL, RDF, XML dan HTML. Protégé menyediakan kemudahan *plug and play* yang membuatnya fleksibel untuk pengembangan *prototype*.

Protégé dibuat dengan menggunakan bahasa Java. Semua alat-alat dalam protégé dapat digunakan melalui *Graphical User Interfece* (GUI) dengan menyediakan *Tab* untuk masing-masing bagian dan fungsi standar. *Class Tab* dalam *editor ontology* berfungsi untuk mendefinisikan *class* dan hirarki *class*, *property* dan nilai *property* tersebut, relasi antara *class* dan *property* dari relasi tersebut (protege.stanford.edu,2005).

Untuk lebih jelasnya tentang komponen *Ontology* dapat dilihat pada gambar 2.6 :



Gambar 2.6 Tampilan Protégé 3.4.5

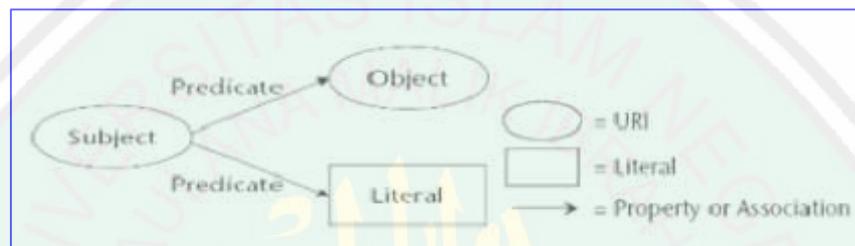
2.5 RDF (*Resource Description Framework*)

2.5.1 Pengertian RDF

RDF (*Resource Description Framework*) merupakan bahasa yang digunakan untuk merepresentasikan metadata. RDF mendukung *interoperabilitas* antar aplikasi yang mempertukarkan informasi yang bersifat *machine-understandable* di web. *Semantic web* terdiri dari data yang ditulis dalam bahasa yang dapat dimengerti oleh mesin, seperti RDF. RDF menggunakan *graf* untuk merepresentasikan kumpulan pernyataan. Simpul dalam *graf* mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas.

RDF merupakan spesifikasi kerangka kerja yang diterbitkan oleh *World Wide Web Consortium* (W3C) sebagai model *metadata* untuk merepresentasikan *resource* yang terdapat di *web* (W3C, 2004). RDF menggunakan istilah tertentu

untuk menguraikan suatu statemen. Secara rinci bagian yang mengidentifikasi didalam statemen dapat dikatakan sebagai *subject*, karakteristik (*property*) dari *subject* disebut sebagai *predicate*, sedangkan nilai dari *property* disebut sebagai *object*, atau dikenal dengan *triple* dalam terminologi RDF.



Gambar 2.7 Triple Graf RDF

(Nurnawati : 2012)

Selain itu RDF memiliki 3 (tiga) obyek tipe:

a. Sumber daya (*Resource*)

Segala sesuatu yang digambarkan dengan RDF disebut resource. Resource ini bisaberupa keseluruhan halaman web, misalnya dokumen HTML <http://www.w3.org/Overview.html>, bisa juga berupa suatu bagian dari sebuah halaman web. Resource ini biasanya diberi nama menggunakan URI (*Uniform ResourceIdentifiers*). Segala sesuatu bisa memiliki sebuah URI, karena URI bersifat bisa diperluas maka URI bisa digunakan sebagai pengenalan bagi berbagai macam entitas.

b. Properti (*Property*)

Properti merupakan aspek, karakteristik, atribut, atau relasi khusus yang digunakan untuk menggambarkan sebuah resource. Setiap properti memiliki

arti khusus, mendefinisikan nilai yang mungkin, tipe resource yang digambarkan, dan relasinya dengan properti lain.

c. Pernyataan (*Statement*)

Suatu *resource* tertentu bersama dengan *properti* dan nilai dari *properti* untuk *resource* tersebut membentuk suatu pernyataan RDF. Ketiga bagian ini disebut *subjek, predikat, dan obyek*, membentuk *RDF triple*.

Agar mudah digunakan bahasa *query* ini memiliki sintaks yang mirip dengan SQL. Bahasa *query* menyediakan fitur untuk menghasilkan himpunan *statement* yang lebih kecil, ataupun untuk membandingkan data. Kemampuan lebih lanjut dari bahasa *query* ini menjadi dasar dari kemampuan *inference*. Bahasa *query* yang saat ini banyak digunakan dan telah menjadi standar *de facto* dinamakan SPARQL (Rakhmad, 2008).

2.5.2 RDF-Gravity

RDF-Gravity adalah tool untuk memvisualisasikan RDF/ OWL Grafik / *Ontology*. Fitur utamanya adalah:

- a. Visualisasi Grafik (*Graph Visualization*) RDF Gravity mendefinisikan paket visualisasi di atas JUNG Grafik API. Paket ini menyediakan fungsionalitas berikut:
 - Penyaji dapat melakukan konfigurasi untuk tepi dan node dari grafik, termasuk bentuk node yang berbeda dan dekorasi tepi dll.
 - Memungkinkan konfigurasi dari node diatas dan penyaji tepi berdasarkan pada jenis keunggulan atau node.

- Memilih beberapa model untuk pemilihan node dan tepi grafik.
- Penanganan aksi pada mouse, misalnya, klik dua kali klik, drag, untuk *node* dan tepi grafik.
- *Zooming*

Untuk *layout* grafik, menggunakan algoritma tata letak yang didukung langsung oleh *Jung* API.

b. Global dan Lokal Filter (memungkinkan tampilan secara spesifik pada grafik). Tool ini memungkinkan user untuk menentukan filter Global dan Lokal terhadap tampilan yang diinginkan pada grafik

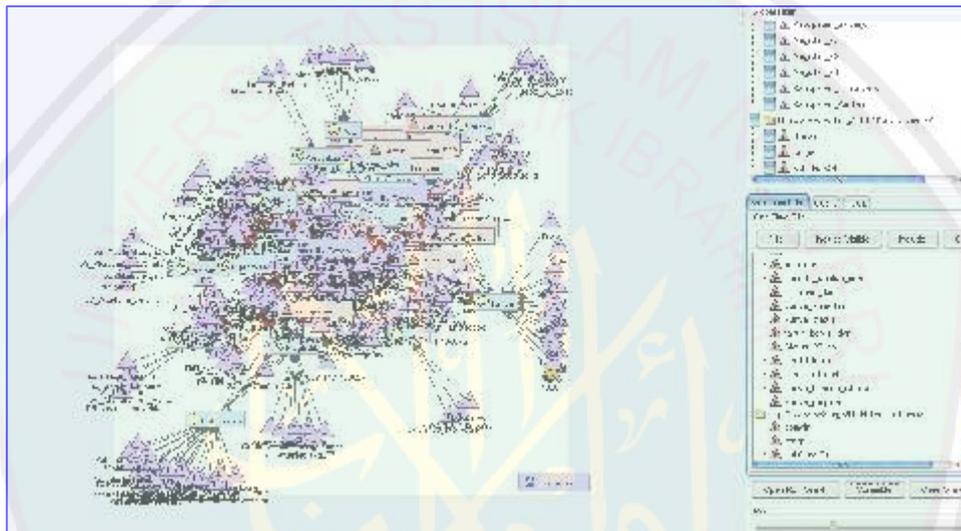
- Filter global memudahkan user untuk menyembunyikan atau melihat tepi spesifik berdasarkan jenis. Misalnya, seseorang dapat menyembunyikan semua sisi dari "rdf: tas", "dc: penulis" atau jenis tertentu yang terjadi dalam *Ontology*.
- Filter Lokal memudahkan user untuk menyembunyikan atau melihat kasus tertentu dari node atau tepi.
- Seorang pengguna dapat mencari konsep dan dapat menambahkan konsep tersebut kedalam grafik untuk divisualisasi.

Fungsionalitas tersebut sangat memungkinkan untuk diterapkan pada struktur grafik yang memiliki kompleksitas lebih tinggi.

c. Pencarian secara penuh (*Full text search*) Gravitasi RDF menyediakan pencarian teks penuh atas konsep-konsep, properti-properti dan *instance* yang ditetapkan dalam sebuah file RDF

- d. Visualising beberapa file RDF RDF Gravity memungkinkan pengguna untuk melihat struktur grafik yang terkandung dalam beberapa file RDF melalui satu tampilan.

Gambar 2.8 berikut ini adalah *screenshot* dari RDF Gravity.



Gambar 2.8 Screenshot RDF Gravity

2.5.3 SPARQL (*Simple Protocol and RDF Query Language*)

SPARQL merupakan bahasa *query* untuk RDF/OWL. Leid Dodds (2005) menyatakan bahwa SPARQL merupakan sebuah bahasa *query* dan protokol akses data, memiliki kemampuan untuk menjadi komponen penting dalam aplikasi Web 2.0: sebagai standar didukung oleh sebuah model data yang fleksibel, dapat memberikan suatu mekanisme permintaan umum untuk semua aplikasi Web 2.0.

Klausa yang digunakan dalam query SPARQL, diantaranya:

a. *PREFIX*

Statement *PREFIX* merupakan sebuah metode yang digunakan sebagai penunjuk yang membawa informasi dalam suatu halaman *web*. Pada dasarnya

PREFIX digunakan untuk menyingkat sebuah *resource*, dalam hal ini dapat diwakili oleh URI (*Uniform Resource Identifier*).

b. *SELECT*

Statement *SELECT* didefinisikan sebuah daftar variabel-variabel yang akan dikembalikan sebagai hasil dari eksekusi *query*. Setiap variabel diawali dengan notasi(?).

c. *WHERE*

Statement *WHERE* didefinisikan sederetan *triple pattern* yang harus dimiliki oleh setiap hasil *query* yang valid. Seluruh pola yang merepresentasikan suatu kalimat RDF harus sesuai dengan RDF *triples*, yaitu terdiri dari subyek, predikat dan obyek. Ketiga RDF *triple* tersebut dapat direpresentasikan oleh URI atau sebuah variable dan literal.

d. *OPTIONAL*

Statement *OPTIONAL* digunakan untuk mengatasi ketidakcocokan struktur *query* dengan pola yang ada pada graf RDF. Contoh *source code query*

SPARQL:

SPARQL:

```
Select * WHERE {9
```

```
?x vcard:hasJudul ?judul . ?x vcard:hasMataKuliah ?matakuliah . ?x vcard:hasDosen ?dosen .
```

```
?x vcard:hasLinkStaff ?linkstaff . ?x vcard:hasLinkView ?linkview . FILTER (?judul =
```

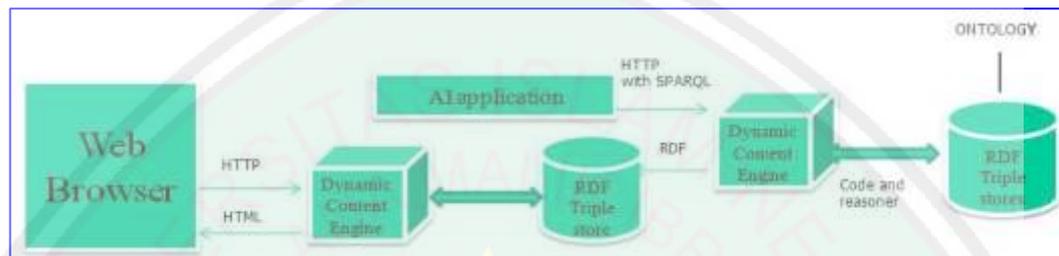
```
“Nama_Judul_Materi_Yang_Dicari” && ?matakuliah = “Nama_Matakuliah_Yang_Dicari”
```

```
&& ?dosen =” Nama_Matakuliah_Yang_Dicari ”)
```

```
}
```

SPARQL mirip dengan SQL dan digunakan untuk akses RDF berbasis dan ekstrak data dari database tradisional, maka menghasilkan hasil yang sempurna.

Gambar 2.9 berikut menjelaskan cara kerja SPARQL:



Gambar 2.9 Query RDF menggunakan SPARQL

(Sumber: Gupta, dkk. 2010)

2.5.4 Jena Framework

Jena adalah *framework* Java yang digunakan untuk membangun aplikasi *semantic web*. Implementasi dari Jena API, termasuk kode sumber yang bekerja untuk semua contoh yang digunakan dalam tutorial ini dapat didownload dari <http://jena.apache.org/download/> yang berbasis *open source* yang menggunakan bahasa Java. *Framework* ini menyediakan lingkungan pemrograman RDF, RDF Schema OWL dan SPARQL. Jena merupakan sebuah mesin *rule-based inference* yang berguna untuk melakukan penalaran berdasarkan OWL dan RDFS *ontology*, serta pengembangan dalam strategi penyimpanan untuk menyimpan N-Triple ke dalam *memory* atau *disk*.

Apache Jena *framework* Java untuk membangun aplikasi *semantic web*. Apache Jena menyediakan sekumpulan *tool* dan *library* java untuk membantu membangun aplikasi *semantic web* dan *linked data*. Jena memiliki kelas

ModelFactory yang dapat digunakan untuk membuat berbagai model. Pada Jena, subjek setiap *statement* selalu berupa sebuah *Resource*, sedangkan predikat direpresentasikan oleh *Property*, dan objek bisa direpresentasikan oleh sebuah *Resource* lain maupun sebuah nilai literal. Jena juga memiliki kemampuan untuk melakukan penalaran (*inference*) terhadap model yang telah dibuat. Dengan kata lain, Jena memiliki kemampuan untuk membuat *statements* tambahan yang belum ditulis secara eksplisit di dalam model (Niko.2007).

2.5.5 Smore (*semantic markup, ontology and rdf editor*)

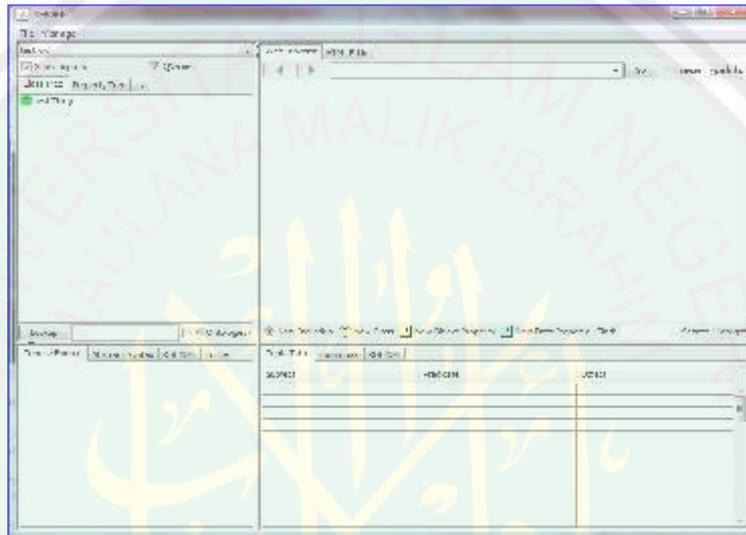
Smore *Markup Semantic, Ontology, dan RDF Editor* adalah sebuah tool yang dikembangkan para peneliti di University of Maryland, *College Park*. Smore meruakan aplikasi yang menggabungkan empat aplikasi yang terpisah menjadi satu untuk memberikan hanya dukungan tersebut. Smore merupakan aplikasi berbasis Java, salah satu produk yang menarik untuk keluar dari *Web Semantic Research Group*, bersama dengan beberapa konverter, RDF modifikasi, dan alat-alat lain yang berguna dan utilitas (<http://etutorials.org/>).

Smore merupakan alat yang digunakan untuk mengintegrasikan pembuatan konten beserta penjelasannya. Hal ini merupakan fasilitas pada *Semantic Web* untuk meng-markup berbagai jenis media seperti foto, html, ataupun e-mail dan dengan begitu memberikan tingkat fleksibilitas menjadi lebih tinggi dalam penggunaan serta modifikasi suatu *ontology* (B. Parsia,dkk. 2005).

Smore diciptakan untuk memudahkan pengguna dalam melakukan *markup* pada *semantic web*. Smore akan terbuka dengan interface yang terdiri dari empat

jendela terpisah dalam satu frame. Setiap jendela berisi aplikasi terpisah: web browser dalam satu, browser web *Ontology* di tempat lain, editor HTML pada bagian ketiga, dan representasi data semantik di keempat.

Berikut gambar 2.10 Interface dari Smore versi 2.6.4 :



Gambar 2.10 Tampilan Smore versi 2.6.4

2.5.6 Sphider

Sphider adalah *web-spider* dan mesin pencari open-source. Ini mencakup crawler otomatis, yang dapat mengikuti link yang ditemukan pada sebuah situs, dan indexer yang membangun indeks dari semua istilah pencarian yang ditemukan di halaman. Sphider ditulis dalam PHP dan menggunakan MySQL sebagai databasenya. Untuk download Sphider di <http://www.sphider.eu/download.php>. Versi yang paling baru adalah versi 1.3.5.

Spider mengerjakan isi setiap halaman lalu dianalisis untuk menentukan cara mengindeksnya (misalnya, kata-kata diambil dari judul, subjudul, atau *field* khusus yang disebut meta tag) menyimpan seluruh atau sebagian halaman sumber

(yang disebut *cache*) maupun informasi tentang halaman web itu sendiri. Sebagai *interface* Spidder dapat dilihat pada gambar 2.11 berikut:



Gambar 2.11 Tampilan Spidder

2.6 Metode MCDM (*Multi Criteria Decision Making*)

Multi Criteria Decision Making (MCDM) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu (Triantaphyllou,dkk.1998). Multi kriteria pengambilan keputusan atau beberapa kriteria analisis keputusan adalah sub disiplin riset operasi yang secara *eksplisit* mempertimbangkan beberapa kriteria dalam lingkungan pengambilan keputusan. Apakah dalam kehidupan sehari-hari atau dalam pengaturan profesional, biasanya ada beberapa kriteria yang bertentangan yang perlu dievaluasi dalam membuat keputusan (Sartin.2006).

Menurut *Ling Xu & Dr. Jian*, 2011 Beberapa kriteria pengambilan keputusan MCDM mengacu pada pembuatan keputusan di hadapan beberapa, biasanya saling bertentangan, kriteria. Masalah MCDM yang umum dalam

kehidupan sehari-hari. Dalam konteks pribadi, rumah atau mobil yang membeli dapat dicirikan dari segi harga, ukuran, gaya, keamanan, kenyamanan, dll. Pembelian departemen perusahaan besar sering perlu untuk mengevaluasi pemasok mereka menggunakan berbagai kriteria di daerah yang berbeda, seperti layanan purna jual, manajemen kualitas, stabilitas keuangan.

Menurut Triantaphyllou, 1997 Louisiana State University ada beberapa kriteria multi metode pengambilan keputusan Ada tiga langkah utama dalam memanfaatkan pengambilan keputusan teknik yang melibatkan analisis numerik dari satu set alternatif diskrit: Pertama, Menentukan kriteria dan alternatif yang relevan. Kedua, Melampirkan langkah numerik untuk kepentingan relatif (bobot) kriteria dan terhadap dampak (ukuran kinerja) dari alternatif dalam hal kriteria ini. Dan yang ketiga adalah, Pengolahan nilai numerik untuk menentukan peringkat masing-masing alternatif.

Keputusan merupakan proses pemilihan alternatif terbaik dari banyak alternatif. Pengambilan keputusan terkadang melibatkan pengalaman. Tidak jarang pula *decision maker* (DM) mengamabil keputusan dengan menggunakan insting atau intuisi, sehingga menghasilkan keputusan yang tepat (Iksan, 2006). Karenanya, untuk menghasilkan yang tepat DM harus memperoleh informasi sebanyak mungkin.

Adapun matrik keputusan pada *Multi Criteria Decision Making* (MCDM) secara umum dapat dilihat pada gambar 2.12 :

	c_1	c_2	c_3	c_m
a_1	v_{11}	v_{12}	v_{13}	v_{1m}
a_2	v_{21}	v_{22}	v_{23}	v_{2m}
a_3	v_{31}	v_{32}	v_{33}	v_{3m}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
a_n	v_{n1}	v_{n2}	v_{n3}	v_{nm}

Gambar 2.12 Matrik keputusan MCDM

Keterangan dari matrik keputusan pada MCDM:

A: Set variabel alternatif keputusan ($a_1, a_2, a_3, \dots, a_n$)

C: Set kriteria keputusan ($c_1, c_2, c_3, \dots, c_m$)

V_{ij} : Nilai alternatif i ($i=1, \dots, n$) dievaluasi dengan kriteria j ($j=1, 2, \dots, m$)

Konsep pembobotan MODM/MCDM dalam tingkat relatif (*numerik*) dapat diturunkan dari skala ordinal (Merupakan skala yang membedakan kategori berdasarkan tingkat atau urutan).

W_I = bobot kriteria I ($I=1, 2, \dots, k$)

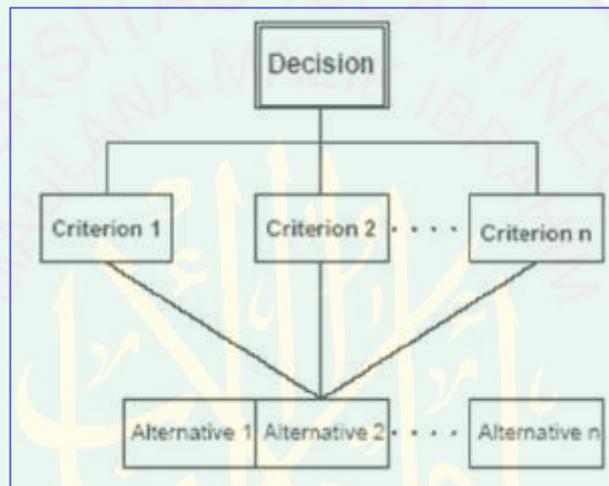
$0 < w_1 < 1$

$$\sum_{I=1}^n W_I = 1$$

Dimana : $w_1 > w_h$: kriteria I lebih penting dibandingkan kriteria h

MCDM selalu melibatkan lebih dari satu kriteria yang saling menimbulkan *trade off* keputusan dimana tingkat kepuasan dari suatu kriteria berakibat pada penurunan kepuasan kriteria lainnya. Dalam menilai tingkat kepentingan dalam *multiple criteria*, ada beberapa metode yang dapat digunakan.

Tingkat keputusan relatif dari suatu kriteria disebut prioritas (*priority*) dan pada bobot (*weight*). Bobot digunakan untuk membedakan tingkat kepentingan dari beberapa kriteria dengan prioritas yang berbeda. Berikut gambar 2.11 *Tree Decision Making*:



Gambar 2.11 Multi *Criteria Decision Making* (MCDM) *Tree*
(Sumber: Kiran, 2012)

Kriteria merupakan ukuran, aturan-aturan ataupun standar-standar yang memandu suatu pengambilan keputusan. Pengambilan keputusan dilakukan melalui pememilihan atau memformulasikan atribut-atribut, obyektif-obyektif, maupun tujuan-tujuan yang berbeda, maka atribut, obyektif maupun tujuan dianggap sebagai kriteria. Kriteria dibangun dari kebutuhan-kebutuhan dasar manusia serta nilai-nilai yang diinginkannya. Ada dua macam kategori dari *Multi-criteria decision making* (MCDM), yaitu :

2.6.1 *Multiple Objective Decision Making* (MODM)

Multiple Objective Decision Making (MODM) menyangkut masalah perancangan (*design*), dimana teknik-teknik matematik optimasi digunakan, untuk jumlah alternatif yang sangat besar (sampai dengan tak berhingga) dan untuk menjawab pertanyaan apa (*what*) dan berapa banyak (*how much*).

2.6.2 *Multiple Attribute Decision Making* (MADM)

Multiple Attribute Decision Making (MADM), menyangkut masalah pemilihan, dimana analisa matematis tidak terlalu banyak dibutuhkan atau dapat digunakan untuk pemilihan hanya terhadap sejumlah kecil alternatif saja. Ada beberapa metode yang sering digunakan dalam teknik MADM, seperti Metode *Weighted Sum Model* (WSM) yang menawarkan cara yang fleksibel dan sederhana kepada pembuat keputusan untuk menganalisis masalah-masalah *Multi-criteria*, pembangkitan rekomendasinya menggunakan teknik *Multiple Attribute Decision Making* (MADM).

Proses pengambilan keputusan dari suatu sistem yang kompleks, pendekatan "*multiple criteria*" digunakan untuk mendeskripsikan situasi keputusan. MODM melibatkan lebih dari satukriteria dengan banyak alternatif, sedangkan MADM merupakan permasalahan pemilihan alternatif terbaik. Perbedaan MODM dan MADM dapat dilihat pada tabel 2.2:

Tabel 2.2 Perbedaan MODM dan MADM
(Sumber: Sartin 2008)

Faktor	Metode Multi Atribut (MADM)	Metode Multi Obyektif (MODM)
Kriteria	Atribut	Obyktif
Obyektif	Implisit	Eksplisit
Atribut	Ekplisit	Implisit
Kendala	Matriks	Aktif
Alternatif	Jumlah terbatas	Jumlah tidak terbatas dan kontinyu
Interaksi	Jarang	Lebih sering
Pemakaian	Problem seleksi dan pemilihan alternatif	Problem konsepsi dan rekayasa

Alasan peneliti mengambil topik pemilihan perguruan tinggi ini selain karena sebagai orang yang bekerja di bidang pendidikan juga karena pemilihan perguruan tinggi ditentukan dengan beberapa kriteria tertentu dengan menentukan nilai bobot yang dinilai oleh badan akreditasi perguruan tinggi nasional (BAN-PT) di seluruh Indonesia, sehingga untuk selanjutnya peneliti ingin memberikan solusi berupa sistem pengambilan keputusan yang bisa digunakan secara efektif dan efisien.

2.6.3 Sifat-sifat MCDM dalam Memilih Kriteria

Sifat-sifat yang harus diperhatikan dalam memilih kriteria pada setiap persoalan pengambilan keputusan adalah sebagai berikut (kusumuadewi.2006):

- a. Operasional, sehingga dapat digunakan dalam analisis. Sifat operasional ini mencakup beberapa pengertian, antara lain adalah bahwa kumpulan kriteria

ini harus mempunyai arti bagi pengambil keputusan, sehingga ia dapat benar-benar menghayati implikasinya terhadap alternatif yang ada.

- b. Lengkap, sehingga dapat mencakup seluruh aspek penting dalam persoalan tersebut. Suatu set kriteria disebut lengkap apabila set ini dapat menunjukkan seberapa jauh seluruh tujuan dapat dicapai.
- c. Tidak berlebihan, sehingga menghindarkan perhitungan berulang. Dalam menentukan set kriteria, jangan sampai terdapat kriteria yang pada dasarnya mengandung pengertian yang sama.
- d. Minimum, agar lebih mengkomprehensifkan persoalan. Dalam menentukan sejumlah kriteria perlu sedapat mungkin mengusahakan agar jumlah kriterianya sesedikit mungkin. Karena semakin banyak kriteria maka semakin sukar pula dalam menghayati persoalan dengan baik, dan jumlah perhitungan yang diperlukan dalam analisis akan meningkat dengan cepat.

2.6.4 *Weighted Product Model (WPM)*

Model produk tertimbang (WPM) adalah multi-kriteria yang populer analisis keputusan (MCDA) / multi-kriteria pengambilan keputusan (MCDM) metode. Hal ini mirip dengan model jumlah tertimbang (WSM). Perbedaan utama adalah bahwa alih-alih penambahan dalam operasi matematika utama sekarang ada adalah perkalian. Seperti metode MCDA / MCDM, adalah himpunan berhingga dari alternatif keputusan yang dijelaskan dalam hal sejumlah kriteria keputusan. Setiap keputusan alternatif dibandingkan dengan orang lain dengan mengalikan sejumlah rasio, satu untuk setiap kriteria keputusan. Setiap rasio

dinaikkan menjadi setara kekuatan untuk bobot relatif dari kriteria yang sesuai (Wang, dkk. 2006).

Seperti semua metode MADM, WPM adalah himpunan berhingga dari alternatif keputusan yang dijelaskan dalam istilah beberapa kriteria keputusan. Vertikal serah terima masalah keputusan dapat dinyatakan sebagai bentuk matriks dan setiap baris i sesuai dengan jaringan kandidat i dan setiap kolom j sesuai dengan atribut. Triantaphyllou, dkk. 2000 dalam penelitiannya WPM ini kadang-kadang disebut analisis berdimensi karena struktur menghilangkan unit measure. Thus, WPM dapat digunakan dalam masalah pengambilan keputusan tunggal dan multi-dimensi. Keuntungan dari metoda yang bukan nilai yang sebenarnya dapat menggunakan yang relatif.

Misalkan masalah MCDA diberikan didefinisikan pada alternatif m dan n kriteria keputusan. Selanjutnya, mari kita asumsikan bahwa semua kriteria kriteria manfaat, yaitu, semakin tinggi nilai-nilai, semakin baik. Selanjutnya misalkan w_j menunjukkan bobot relatif pentingnya kriteria C_j dan a_{ij} adalah nilai kinerja alternatif A_i ketika dievaluasi dalam hal kriteria C_j . Kemudian, jika seseorang ingin membandingkan dua alternatif A_K dan A_L (di mana $m \geq K, L \geq 1$) maka, produk berikut harus dihitung.

$$P(A_K/A_L) = \prod_{j=1}^n (a_{Kj}/a_{Lj})^{w_j}, \text{ for } K, L = 1, 2, 3, \dots, m.$$

Jika rasio $P(A_K/A_L)$ lebih besar atau sama dengan nilai 1, maka menunjukkan bahwa alternatif A_K lebih diinginkan daripada alternatif A_L (dalam

kasus maksimalisasi). Jika kita tertarik dalam menentukan alternatif terbaik, maka alternatif terbaik adalah salah satu yang lebih baik dari atau setidaknya sama dengan semua alternatif lainnya.

2.6.5 Memilih Perguruan Tinggi

Proses pengambilan keputusan meliputi penetapan tujuan, pembatasan dan analisis masalah, pencarian alternatif, pemilihan alternatif yang maksimal, pelaksanaan keputusan, serta penilaian dan monitoring. Proses pengambilan keputusan perlu dimengerti oleh lulusan SMA sewaktu akan memasuki perguruan tinggi, karena melalui tahap-tahap tersebut keputusan yang akan diambilnya akan lebih efektif (Ali, 2006).

Sebelum calon mahasiswa memilih perguruan tinggi untuk dimasuki, biasanya mereka memiliki beberapa faktor yang dijadikan dasar pertimbangan dalam mengambil keputusan. Pada umumnya kriteria yang dipilih meliputi status akreditasi, citra, fasilitas fisik, biaya, mutu dosen, mutu lulusan, prospek, dan sebagainya. Adapun faktor memilih perguruan tinggi diantaranya :

a. Minat

Faktor utama yang harus anda pertimbangkan adalah minat. Hampir boleh dipastikan, tidak ada mahasiswa yang berhasil dalam studinya jika itu bertentangan dengan minatnya.

b. Biaya

Kemampuan keuangan sangat menentukan pilihan anda. Ini adalah faktor terpenting berikutnya yang harus anda perhitungkan. Kuliah di perguruan

tinggi melibatkan banyak komponen biaya, mulai dari pendaftaran, biaya praktikum, biaya KKN, dan sebagainya.

c. Status Akreditasi

Status akreditasi menunjukkan mutu atau kinerja suatu perguruan tinggi dalam menyelenggarakan suatu program studi. Status ini diberikan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) setelah diadakan penilaian tentang semua unsur yang diperlukan, seperti fasilitas pendidikan, perbandingan dosen tetap dan mahasiswa, kurikulum pendidikan, dan sebagainya.

d. Jalur dan Jenjang Pendidikan

Pendidikan tinggi di Indonesia mengenal dua jalur pendidikan, yaitu jalur akademik (jenjang sarjana) dan jalur profesional (jenjang diploma). Jalur akademik menekankan pada penguasaan ilmu pengetahuan, sedangkan jalur profesional menekankan pada penerapan keahlian tertentu.

e. Fasilitas Pendidikan

Gedung megah dan ber-AC saja tidak cukup untuk menjamin berlangsungnya proses pembelajaran yang baik. Fasilitas pendidikan pada suatu perguruan tinggi lebih pada ketersediaan dan kelengkapan laboratorium (komputer, akuntansi, bahasa, dan lain-lain), bengkel, studio, dan perpustakaan sangat diperlukan untuk menunjang keberhasilan mahasiswa.

2.7 Kolerasi Rekomendasi Perguruan Tinggi Dalam Perspektif Al-Qur'an dan Hadist

Pendidikan secara sederhana dapat diartikan sebagai usaha manusia untuk membentuk atau membina kepribadian sesuai dengan nilai-nilai yang tepat didalam masyarakat. Rasulullah telah bersungguh-sungguh mendidik sahabat dan generasi muslim, hingga mereka memiliki kesempurnaan Akhlak dan ilmu pengetahuan. Dari latar belakang di bab I Nabi yang Mulia, Nabi Muhammad Saw bersabda bahwa *“sebaik-baik manusia adalah yang paling banyak memberikan manfaat kepada sesama”*.

Sistem yang mempermudah proses pencarian informasi perguruan tinggi sangat dibutuhkan disetiap kalangan. Dengan semakin berkembangnya kemajuan zaman terutama dibidang ilmu teknologi informasi, maka informasi yang akurat merupakan suatu kebutuhan utama bagi masyarakat khususnya didalam bidang ilmu pendidikan. Terutama kemajuan di bidang perguruan tinggi yang mampu memberikan informasi yang handal, cepat, dan akurat yang dibutuhkan dalam pengambilan suatu keputusan. Hal ini dengan adanya sistem ini dibuat untuk membantu mempercepat, mempermudah dan lebih akurat dalam dalam merekomendasikan perguruan tinggi yang dimaksud.

يُرِيدُ اللَّهُ بِكُمُ الْيُسْرَ وَلَا يُرِيدُ بِكُمُ الْعُسْرَ

Artinya: "Allah menghendaki kemudahan bagimu, dan tidak menghendaki kesukaran bagimu." (QS. Al-Baqarah: 185)

Mudah dan mempermudah merupakan salah satu dari karakteristik Islam.

Islam ingin mempermudah umatnya dalam menerima dan menjalankan kebenaran

dan tidak ingin mempersulit mereka. Sehingga orang menerima dan menjalankan Islam dengan lapang dada dan senang. Hal ini diperkuat oleh Al-Qur'an Di dalam Al-Qur'an kita temukan, Allah menyatakan bahwa islam yang disyariatkan kepada hamba-hamba-Nya bukan untuk menyulitkan mereka. Sebagaimana firman Allah:

وَجَاهِدُوا فِي اللَّهِ حَقَّ جِهَادِهِ ۗ هُوَ اجْتَبَاكُمْ وَمَا جَعَلَ عَلَيْكُمْ فِي الدِّينِ مِنْ حَرَجٍ



Artinya : *“Dan berjihadlah kamu pada jalan Allah dengan Jihad yang sebenarnya. Dia telah memilih kamu dan Dia sekali-kali tidak menjadikan untuk kamu dalam agama suatu kesempitan”.* (QS. Al-Hajj: 78)

Maksud dari ayat diatas adalah jika ingin beruntung, banyak-banyaklah berbuat baik. Tundukkan hatimu, di hadapan Allah Sang Penguasa alam semesta, yang Maha Mengatur segala sesuatunya. Ingat-ingatlah bahwa Allah telah memilih kita sebagai makhluk ciptaannya yang sempurna, dan tidak menjadikan kita sulit di dalam mempelajari ilmu agama sepanjang kita benar-benar berniat. Artinya hal-hal yang membuat kalian sulit untuk melakukannya, untuk itu Dia memberikan kemudahan kepada kalian (Tafsir Jalalain versi digital).

Apabila kita mengetahui bahwa sebenarnya kita mampu berbuat sesuatu untuk menolong kesulitan orang lain, maka segeralah lakukan, segeralah beri pertolongan. Terlebih lagi bila orang itu telah memintanya kepada kita. Karena

pertolongan yang kita berikan, akan sangat berarti bagi orang yang sedang kesulitan. Sebagaimana dari Anas r.a berkata Nabi Muhammad bersabda :

يَسِّرُوا ,

Artinya: *Ringankanlah ajaran da'wahmu dan jangan mempersukar, dan bergembiralah pengikutmu dan jangan kamu gusarkan.*” (Bukhari, Muslim)

Dari hadist diatas Rasulullah bersabda (janganlah mempersulit)

dengan maksud untuk mengingatkan, bahwa memberikan kemudahan kepada orang lain harus selalu dilakukan dalam setiap situasi dan kondisi. Dengan demikian pula dengan sabda Nabi, setelah kata .

(dan berilah berita gembira). Menggunakan metode bertahap dalam megajarkan suatu ilmu,tentunya sistem rekomendasi perguruan tinggi jatim sangat membantu dan mempermudah dalam penggunaannya (menguasainya).

Sebagai manusia yang dikaruniai akal, manusia diperintahkan untuk selalu berfikir dan mencari sesuatu yang belum kita ketahui manfaat. Begitu juga bagi para pemikir (orang-orang yang berfikir, berilmu, berakal) Allah memberikan berita gembira terhadap mereka melalui beberapa firman-Nya. Sebagaimana dalam firman Allah SWT:

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَأَخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِّأُولِي الْأَلْبَابِ ﴿١٩٠﴾

Artinya: *“Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal”.* (QS. Ali Imran: 190)

Ayat diatas menyeru kepada kita untuk berfikir bahwa semua yang diciptakan oleh Allah memiliki maksud dan tujuan. Sains dan teknologi yang dikembangkan dari masa ke masa yang dapat dimanfaatkan manusia manusia. Dari ayat tersebut juga mengindikasikan bahwa sebenarnya sains dan teknologi yang lahir dari proses berfikir tidak lain adalah untuk membuktikan kebenaran akan firman-firman Allah SWT. Hakekatnya ada pemahaman aktivitas mulia yang diperintahkan untuk kita semua yaitu berfikir untuk memberi manfaat. Membaca informasi serta mencari sesiatu hal yang belum diketahui di sekitar kita.

Apabila kita mengetahui bahwa sebenarnya kita mampu berbuat sesuatu untuk menolong kesulitan orang lain, maka segeralah lakukan, segeralah beri pertolongan. Dari Abu Hurairah ra, Nabi SAW, bersabda: *“Barang siapa yang melepaskan satu kesusahan seorang mukmin, pasti Allah akan melepaskan darinya satu kesusahan pada hari kiamat. Barang siapa yang menjadikan mudah urusan orang lain, pasti Allah akan memudahkannya di dunia dan di akhirat. Barang siapa yang menutupi aib seorang muslim, pasti Allah akan menutupi aibnya di dunia dan di akhirat. Allah senantiasa menolong hamba Nya selama hamba Nya itu suka menolong saudaranya”*. (HR. Muslim, lihat juga Kumpulan Hadits Arba'in An Nawawi hadits ke 36).

Dari penjelasan hadits diatas dapat disimpulkan. Disunnahkan bahkan Allah sangat senang kepada hambanya apabila senantiasa membantu dan memberi kemudahan kepada sesama muslim. Banyak dan bermacam-macam bentuk dari kebutuhan. Semua yang namanya manusia hidup didunia tidak lepas akan bertemu yang namanya kesulitan demi kesulitan yang memerlukan bantuan dari orang lain. Allah menciptakan sedemikian rupa agar manusia yang

mempunyai kemampuan dan kelebihan baik berupa harta maupun yang lain itu agar mereka mau berpikir dan mempunyai kepedulian antar sesama.

Informasi perguruan tinggi yang dicari tentang sisi keunggulan dan kekurangannya masing-masing pasti saling berkesinambungan dengan kata lain tidak ada perguruan tinggi yang sempurna. Sebuah kampus pun mungkin mempunyai keunggulan di program studi atau di aspek tertentu, namun kurang unggul di program studi atau aspek yang lain. Idealnya, cari sebuah kampus yang lebih banyak keunggulannya daripada kelemahannya. Di sinilah informasi yang menyeluruh perlu dijadikan acuan sebelum memilih perguruan tinggi. Memilih perguruan tinggi yang tepat merupakan sebuah keputusan penting bagi setiap peminat perguruan tinggi, karena akan menentukan masa depan dan karir mereka, terlebih jika keputusan itu sudah mengarah pada pemilihan program studi yang benar-benar diminati.

Penelitian ini, tujuan akhir yang diraih adalah mempermudah aktivitas untuk memperoleh informasi secara tepat dan akurat dari perguruan tinggi yang diminati. Dalam hal ini penelitian yang dilakukan dengan menerapkan teknologi *semantic web* pada aplikasi rekomendasi perguruan tinggi dengan menggunakan metode banyak kriteria pemilihan. Harapannya dengan penelitian ini, manusia akan mendapatkan kemudahan dalam mencari informasi perguruan tinggi Jawa Timur. Sehingga bisa memilih dan menentukan perguruan tinggi yang diinginkan dengan mudah dan *efesien*.

BAB III

ANALISA DAN PERANCANGAN

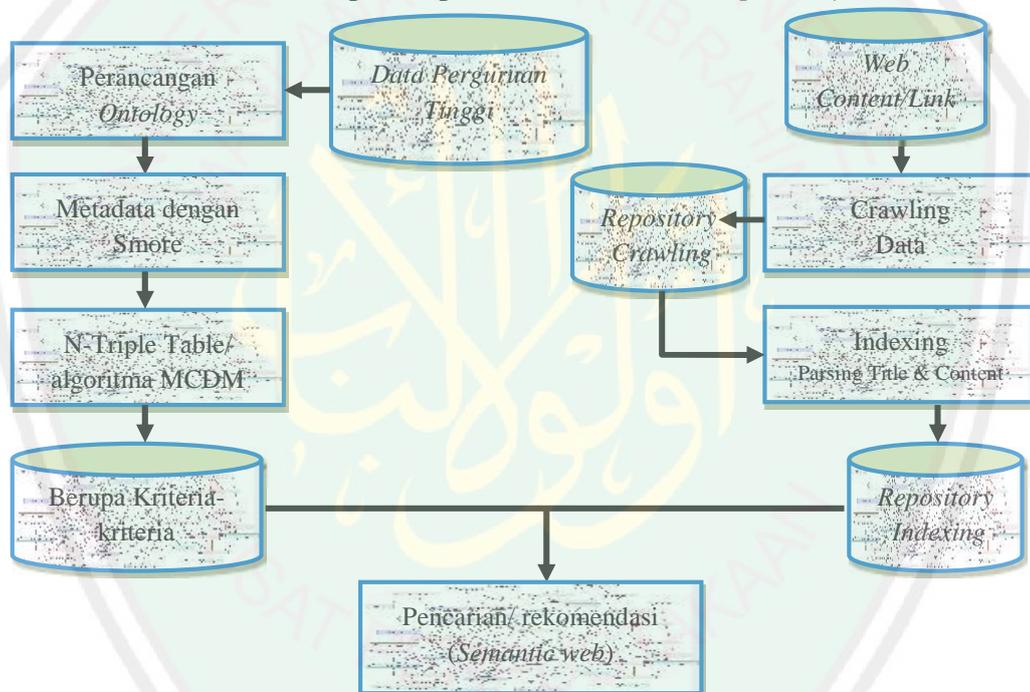
Pada segmen ini perancangan dibagi menjadi dua yaitu, perancangan ontologi dan perancangan sistem. Perancangan ontologi untuk merepresentasikan basis pengetahuan, sedangkan rancangan sistem merupakan gambaran atau perancangan dari sebuah sistem pada perangkat lunak yang terdiri dari rancangan proses, perancangan *flowchart* dan rancangan antar muka.

3.1 Tahapan Proses Pembuatan Sistem dengan *Semantic Web*

Untuk membangun sebuah *semantic web* dengan pemodelan *ontology* diperlukan beberapa tahapan proses serta *tools* sebagai aplikasi pendukung dalam proses pembangunannya, yaitu:

- a. Menyiapkan Data Perguruan Tinggi baik Negeri maupun Swasta Provinsi Jawa Timur.
- b. Perancangan *Ontology*, merancang struktur hirarki serta komponen-komponen yang akan dibutuhkan dalam pembentukan *ontology*. Aplikasi yang digunakan adalah Protégé 3.4.5
- c. Membuat Metadata, Metadata yang dimaksud adalah data *ontology* berupa OWL (*Ontology Web Language*) yang dibentuk dengan pola subjek-predikat-objek. Untuk membuat metadata dibutuhkan *tool* seperti Smore.
- d. Membuat N-Triple *table*, mengkonversikan metadata dari bentuk OWL ke bentuk tabel dalam *database* SQL dengan pola yang sama. Pembuatan N-triple *table* menggunakan Jena yang merupakan *library* java.

- e. *Crawling Data*, tahapan ini akan meng-*crawling* konten dokumen web dari domain web yang telah dibuat. Proses *crawling* menggunakan *tool* Spider.
- f. *Indexing Data*, selanjutnya dilakukan proses *indexing* yang akan melakukan *parsing* data untuk mencari *count title* dan *count content* dari masing-masing dokumen web tersebut.
- g. Pencarian, setelah dengan kriteria-kriteria dan *Repository Indexing* terbentuk maka akan dilakukan proses pencarian dari kedua *repository* tersebut.



Gambar 3.1 Tahapan proses dalam membangun *semantic web*

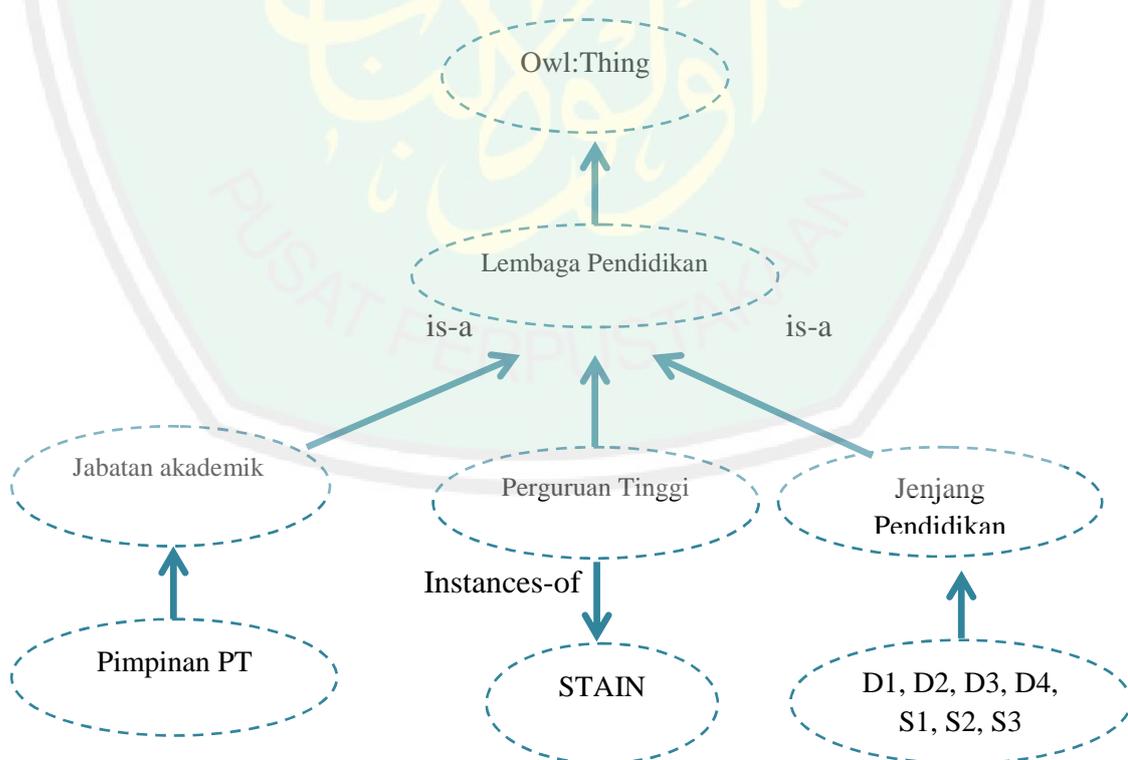
3.2 Pemodelan Data Semantik dengan *Ontology*

Tahapan paling awal dalam membuat aplikasi semantic web adalah dengan membuat sebuah model data semantik yang diwujudkan dalam bentuk ontologi. Dengan ontologi, manusia dan mesin pada hakikatnya telah membuat sebuah kesepakatan antara ke-dua nya untuk memahami suatu

simbol. Simbol tersebut diistilahkan dengan *term*. *Term* inilah yang nanti akan diinterpretasikan oleh mesin dan manusia.

Bagi manusia, arti dari *term* direpresentasikan oleh *term* itu sendiri yang biasanya suatu kata dalam bahasa alami dan relasi antara *term* tersebut yang dapat dipahami oleh manusia. Contoh relasi yang dapat dipahami oleh manusia adalah relasi *class/ subclass* (ditandai dengan *term is-a*). Relasi ini menunjukkan fakta bahwa suatu *class* (*super concept*) adalah lebih umum daripada yang lain (*subclass*). Sebagai contoh, *class* koleksi lebih umum dari pada *class* lembaga pendidikan.

Gambar 3.2 menunjukkan contoh hirarki *is-a* (*taxonomy*) dimana *class* yang lebih umum berada diatas *class* yang lebih khusus.



Gambar 3.2 Hirarki *Is-a*

Dari hirarki tersebut secara sederhana dapat dipahami untuk memperoleh kesimpulan-kesimpulan. Misalnya setiap Perguruan Tinggi (*subclass*) adalah Lemabaga Pendidikan (*class*). *class* menjelaskann satu set obyek. Sebagai contoh, class Perguruan Tinggi bertujuan untuk *men-capture* semua Perguruan Tinggi yang ada. Salah satu contoh adalah STAIN (seperti yang dimodelkan pada gambar 3.1) mempunyai relasi *instance-of* ke *class* Perguruan Tinggi. Relasi *instance-of* mempunyai arti bahwa obyek aktual di *capture* oleh *class* Perguruan Tinggi. Karena relasi is-a dinyatakan secara *eksplisit* dan format antara *class* Perguruan Tinggi, Koleksi. STAIN juga harus menjadi *instance* dari *class* Perguruan Tinggi

Hal yang dilakukan sebenarnya adalah menspesifikasikan relasi dengan cara mengkodekan apa yang diketahui manusia secara implisit (seperti manusia mengetahui bahwa setiap perguruan tinggi masuk kedalam kategori lembaga pendidikan) ke dalam bentuk formal eksplisit sehingga akan dapat dimengerti oleh mesin. Dalam hal ini mesin tidaklah benar-benar mengetahui seperti halnya manusia, namun pengetahuan manusia dikodekan sehingga mesin dapat memprosesnya dan dapat menarik kesimpulan seperti halnya yang dilakukan oleh mesin.

Relasi tersebut mungkin dapat mudah dipahami oleh manusia namun tidak bagi mesin. Dengan adanya pemaknaan (semantik) yang dibangun dengan *ontology* maka membuat mesin menjadi dapat lebih memahami seperti yang dipahami oleh manusia. Hal tersebut dikarenakan adanya pengspesifikasian relasi dengan cara mengkodekan apa yang diketahui manusia secara implisit ke dalam bentuk yang eksplisit sehingga akan dapat dimengerti oleh mesin.

Model data semantik seperti yang telah dijelaskan difokuskan pada hubungan antara *entitas*. Semantik data lebih disebutkan sebagai pemodelan berorientasi *property* dari pada berorientasi *obyek*. Sehingga *entitas* semantik bukan diciptakan dari *class* tetapi entitas lebih dimengerti sebagai anggota dari suatu *class* karena properti dari *class* tersebut.

Lembaga pendidikan dari *resource* yang menggunakan properti disebut dengan domain dari *property*. Secara bersamaan, definisi properti mungkin juga mengindikasikan tipe nilai yang dapat diambil oleh *property* tidak mendefinisikan domainnya, maka tidak dapat dilakukan inference terhadap *resource* yang menggunakan *property* tersebut. Demikian juga jika suatu *property* tidak mendefinisikan range, maka tidak dapat dilakukan inference terhadap *resource* yang menggunakan *property* tersebut.

3.3 Perancangan *Ontology*

Perancangan ontologi memeberikan informasi mengani tahapan-tahapan dalam pembangunan ontologi dan menjelaskan mengenai komponen apa saja yang dibutuhkan dalam penggambaran sebuah informasi. Untuk merancang sebuah ontologi, ada beberapa tahapan yang dilakukan peneliti. Tahapan-tahapan tersebut antara lain dijelaskan pada gambar 3.3:



Gambar 3.3 Tahapan Pemodelan *Ontology*

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

3.3.1 Penentuan Konsep dan Domain

Penentuan konsep dan domain pada dasarnya merepresentasikan koleksi semua dokumen yang dilengkapi dengan informasi dan disusklasifikasi dan dikelompokkan kedalam jenis penentuan konsep dan domain pada dasarnya merepresentasikan koleksi semua dokumen yang dilengkapi dengan informasi dan disusun berdasarkan klasifikasi dan dikelompokkan kedalam jenis-jenis yang sama (Class Penentuan konsep dan domain pada dasarnya merepresentasikan koleksi berdasarkan suatu Class).

Tahapan awal dalam membangun sebuah struktur *ontology* yaitu dengan menentukan domain serta ruang lingkungannya. Penentuan konsep dan domain pada dasarnya merepresentasikan koleksi semua dokumen yang dilengkapi dengan informasi dan disusun berdasar suatu klasifikasi dan dikelompokkan kedalam jenis-jenis yang sama (Class).

Domain *Ontology* yang akan dibangun yaitu Informasi perguruan tinggi. *Ontology* ini digunakan untuk aplikasi yang memberikan informasi mengenai Lembaga Pendidikan Tinggi, khusus di daerah Provinsi Jawa Timur. Untuk melengkapi ruang lingkup domain tersebut dibutuhkan beberapa dokumen-dokumen meliputi:

- *Class* Akreditasi,
- *Class* Alamat,
- *Class* Fasilitas,
- *Class* Jabatan_Akademik (mempunyai *subclass* Pimpinan_Perguruan_Tinggi)
- *Class* Kabupaten,
- *Class* Kota,
- *Class* Lembaga_Pendidikan (mempunyai *Subclass* PerguruanTinggi)
- *Class* Periode,
- *Class* Status_Perguruan_Tinggi,
- *Class* Tahun.

3.3.2 Penentuan Daftar Terminologi

Penentuan daftar terminologi menegaskan hal-hal yang berkaitan dengan istilah yang digunakan didalam membuat statement sekaligus memberikan jawaban dari statement yang dibuat sebelumnya. Pada tahap ini, terdapat *enumerasi* kata yaitu menuliskan semua kata yang diperlukan dalam pembentukan *ontology*. Jenis kata yang digunakan yaitu kata benda (untuk membentuk nama *class*) dan kata kerja (untuk membentuk nama *property*).

3.3.3 Definisi Kelas Dan Struktur Hirarki

Representasi definisi kelas dan hirarki kelas adalah mengelompokkan kelas-kelas dengan karakteristik yang sama yang muncul didalam sebuah domain. Uschold dan Gruninger (1996) menegaskan didalam makalah “ *Ontologies: Principles, Methods and Applications*” bahwa ada beberapa pendekatan metode yang dapat digunakan untuk membangun struktur hirarki kelas diantaranya yaitu metode *top-down*, *bottom-up*, *combination*. Untuk aplikasi rekomendasi perguruan tinggi ini digunakan pendekatan *top-down*, dimana kelas-kelas didefinisikan dari mulai konsep yang paling umum sampai konsep yang lebih spesifik. Rancangan kelas hirarki dapat dijelaskan pada gambar 3.4 :



Gambar 3.4 Rancangan Struktur *Hirarki* Kelas Utama

- *Akreditasi*, pengelompokan ringkasan dari akreditasi perguruan tinggi.
- *Alamat*, pengelompokan alamat dari perguruan tinggi.
- *Fasilitas*, pengelompokan dari fasilitas yang dimiliki perguruan tinggi.
- *Jabatan akademik*, pengelompokan pimpinan dari perguruan tinggi.
- *Jenjang pendidikan*, pengelompokan jenjang pendidikan yang ditampilkan.
- *Kabupaten*, pengelompokan letak kabupaten dari perguruan tinggi.
- *Kota*, pengelompokan kota perguruan tinggi
- *Lembaga pendidikan*, pengelompokan perguruan tinggi yang akan ditampilkan
- *Periode*, pengelompokan periode jabatan akademik
- *Tahun*, pengelompokan tahun berdiri dari perguruan tinggi.

Tahap implementasi selanjutnya adalah penentuan *subclass* yang berfungsi untuk mengkhususkan *class* umum. *Subclass* tersebut adalah sebagai berikut seperti yang ditunjukkan pada gambar 3.5 :

Pendidikan. Dengan kata lain, *class* Perguruan Tinggi merepresentasikan sebuah konsep yang merupakan “bagian dari” *class* Lembaga Pendidikan.

3.3.4 Mendefinisikan Property dari Kelas (*Slot*)

Setelah mendefinisikan kelas, maka langkah berikutnya adalah struktur internal dalam konsep harus diperjelas. Untuk merepresentasikan properti dan *slot* dari kelas yaitu dengan mengisikan nilai *slot* yang meliputi : *name*, *type*, *cardinality* dan *other facets*. Berikut *properties* dari kelas-kelas yang telah didefinisikan pada tabel 3.1 sampai dengan tabel 3.10:

Tabel 3.1: Rancangan *Class* yang terbentuk pada *class* Akreditasi

Property	Range	Allowed Values	Type
akreditasi_dari	Single String	Instances	Datatype Property
Punya_akreditasi	PerguruanTinggi	Instances	Object Property

Tabel 3.2: Rancangan *Class* yang terbentuk pada *class* Alamat

Property	Range	Allowed Values	Type
Alamat_dari	Single String	Instances	Datatype Property
punya_alamat	PerguruanTinggi	Instances	Object Property

Tabel 3.3: Rancangan *Class* yang terbentuk pada *class* Kabupaten

Property	Range	Allowed Values	Type
Berada_di_kabupaten	Single String	Instances	Datatype Property
kabupaten_tempat_berdiri	PerguruanTinggi	Instances	Object Property

Tabel 3.4: Rancangan *Class* yang terbentuk pada *class* Tahun

Property	Range	Allowed Values	Type
berdiri_tahun	Single String	Instances	Datatype Property
Tahun_berdiri_dari	PerguruanTinggi	Instances	Object Property

Tabel 3.5: Rancangan *Class* yang terbentuk pada *class* Jenjang_Pendidikan

Property	Range	Allowed Values	Type
Jenjang_pendidikan_dari	Single String	Instances	Datatype Property
Punya_jenjang_pendidikan	PerguruanTinggi	Instances	Object Property

Tabel 3.6: Rancangan *Class* yang terbentuk pada *subclass* jabatan_akademik *class* Pimpinan_perguruan_tinggi

Property	Range	Allowed Values	Type
pimpinan_dari	Single String	Instances	Datatype Property
Punya_pimpinan	PerguruanTinggi	Instances	Object Property

Tabel 3.7: Rancangan *Class* yang terbentuk pada *class* Status

Property	Range	Allowed Values	Type
Status_PT_dari	Single String	Instances	Datatype Property
Punya_status	PerguruanTinggi	Instances	Object Property

Tabel 3.8: Rancangan *Class* yang terbentuk pada *class* Kota

Property	Range	Allowed Values	Type
kota_dari	Single String	Instances	Datatype Property
Punya_kota	PerguruanTinggi	Instances	Object Property

Tabel 3.9: Rancangan *Class* yang terbentuk pada *class* Periode

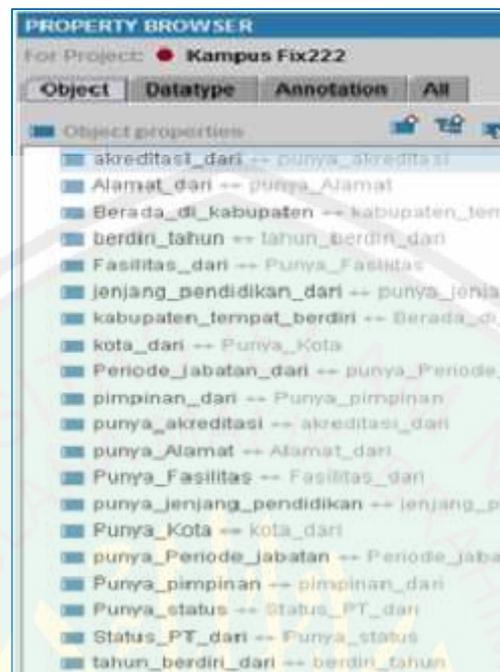
Property	Range	Allowed Values	Type
Periode_jabatan_dari	Single String	Instances	Datatype Property
punya_Periode_jabatan	PerguruanTinggi	Instances	Object Property

Tabel 3.10: Rancangan *Class* yang terbentuk pada *subclass* Lembaga_Pendidikan *Class* PerguruanTinggi

Property	Range	Allowed Values	Type
namaPerguruanTinggi	Single String	Akademi, Institut, Politeknik, Sekolah_Tinggi, Universitas	Datatype Property
Berada_di_kabupaten	Kabupaten	Instances	Object Property
punyaFasilitas	Fasilitas	Instances	Object Property
Jenjang_pendidikan_dari	Multiple Diploma1, Diploma2, Diploma3, Diploma4, Sarjana, Magister, Doktoral	Instances	Object Property

Dari model perancangan *slot* untuk membuat perancangan *class* dengan menambahkan *property-property* di dalamnya sesungguhnya melalui Protégé 3.4.5. Tepatnya pada tab Properties yang di dalamnya juga terdapat beberapa sub tab yaitu tab *owl Classes*, *Object Properties*, *Individuals*, dan *From*. Masing-masing *subtab* tersebut memiliki fungsi dan tujuan berbeda-beda. Tab *Object* digunakan untuk membuat *object properties*, tab *datatype* digunakan untuk membuat *datatype properties*, tab *annotation* digunakan untuk membuat *annotation properties*, sedangkan tab *all* digunakan membuat semua jenis *property* yang terintegrasi dalam satu jendela.

Contoh hasil dari pembuatan *property* pada *class* Lembaga Pendidikan dapat dilihat pada gambar 3.6 :



Gambar 3.6 *Object Properties*

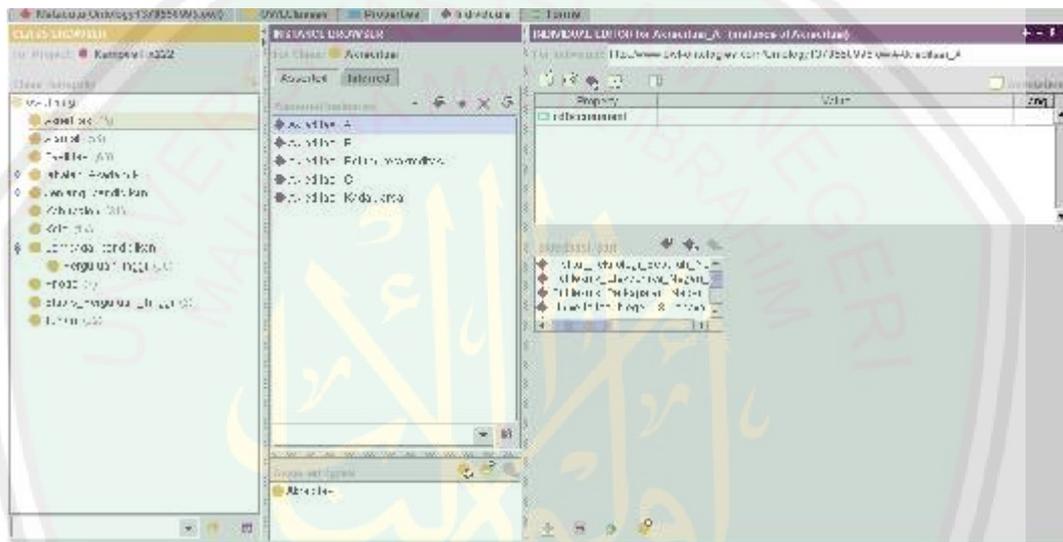
Setelah membuat *property-property*, maka selanjutnya mendefinisikan *Inverse property*. Sesuai dengan namanya, *inverse property* digunakan untuk mendefinisikan *property-property* yang bertolak belakang atau berlawanan. Pada class Akreditasi terdapat *property* yang menggunakan fungsi *inverse* yaitu, *property* “akreditasi_dari” yang memiliki inverse “punya_akreditasi” pada class Akreditasi. Dengan adanya fungsi *inverse property* maka nilai-nilai dari suatu *property* akan secara otomatis terisi ketika *inverse* dari *property* diberi suatu nilai.

3.3.5 Membuat *Instance-Instance*

Setelah pembuatan properti dan *slot* pada setiap kelas yang ada, maka penulis melakukan implementasi *instant* yaitu pengisian nilai-nilai pada *slot* di setiap kelas. Sebelum pembuatan *instant*, penulis *men-design* tampilan form sehingga mempermudah dalam penginputan *instant*.

Instance merupakan komponen yang penting dalam *semantic web*, karena dengan instance aplikasi *semantic* akan memiliki *value* yang nanti ditampilkan pada saat diquery dengan SPARQL.

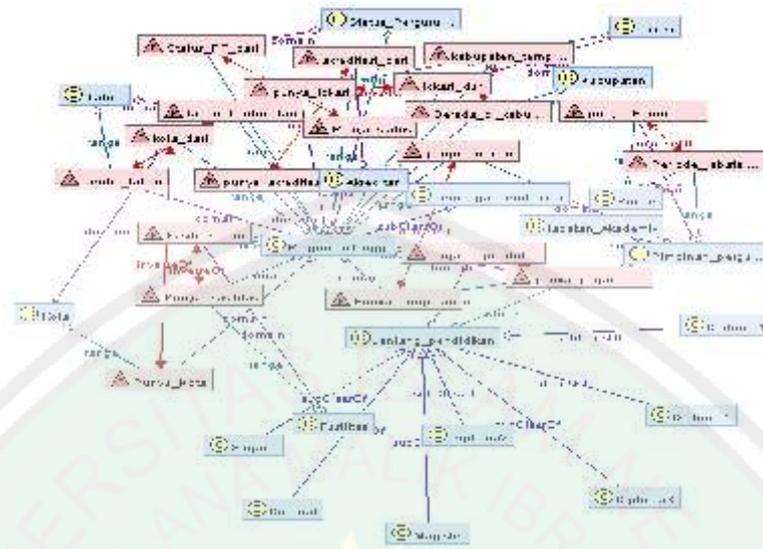
Gambar 3.7 berikut ini contoh instance “Akreditasi_A” disertai dengan nilai dari *individual* instance tersebut.



Gambar 3.7 Instance Akreditasi_A dari Universitas

3.3.6 Visualisasi Graph Ontology RDF Gravity

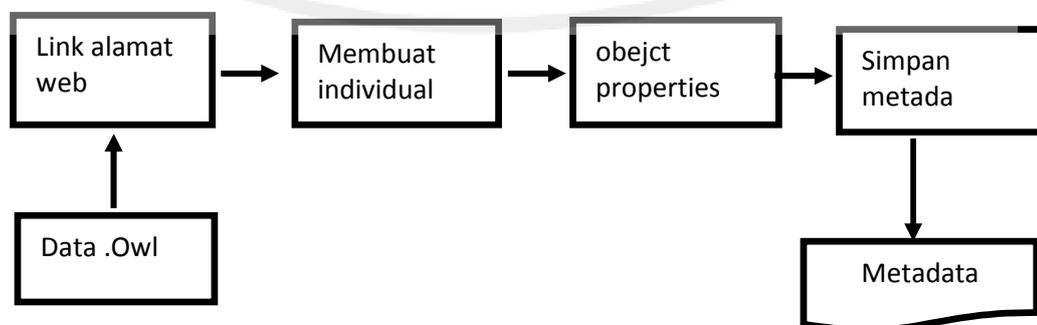
Untuk memastikan relasi yang telah dibuat sesuai dengan tahap perancangan yaitu ketika tahap perancangan ontologi, maka data semantik yang telah dibuat dapat diskematisasikan dengan RDF-Gravity. Sehingga dapat dari ontologi beserta dengan relasi yang dibentuk. Berikut adalah hasil *skematisasi* data semantik terdapat pada gambar 3.8 :



Gambar 3.8 Skematisasi RDF-Gravity

3.4 Membuat Metadata Menggunakan Smore

Setelah *ontology* terbentuk, data pada komponen masih tidak dapat digunakan secara langsung. File *ontology* yang dihasilkan oleh Protégé berupa file metadata (.owl) dan untuk merangkai *ontology* dengan dokumen web terkait perlu dilakukan modifikasi agar dokumen web dapat ter-*markup* pada metadata yang digunakan. Untuk memodifikasi file OWL dapat menggunakan *tool* Smore. Smore dapat meng-*markup* dokumen web untuk membuat metadata yang dibentuk dari *ontology* yang telah dibuat.

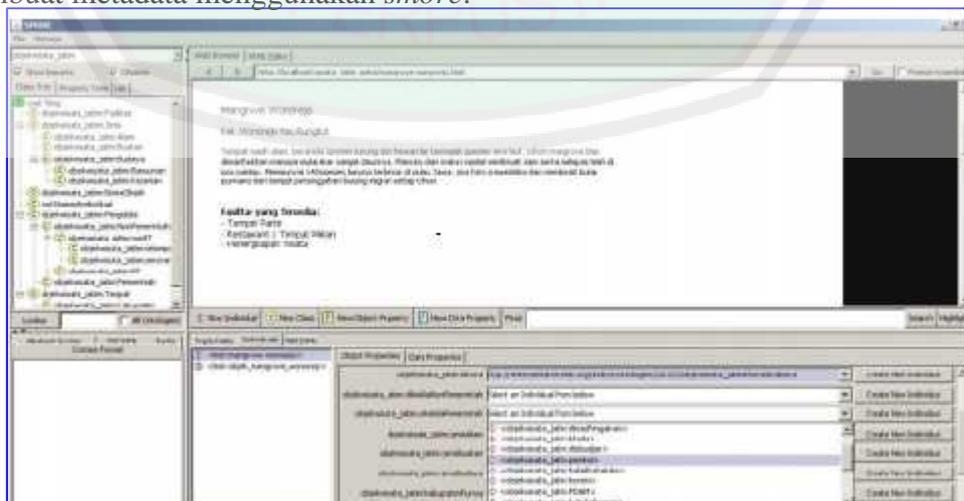


Gambar 3.9 alur yang dilakukan dalam smore

Langkah yang harus dilakukan yang dilakukan pada Smore adalah seperti pada gambar 3.9 :

- Load data ontology* hasil dari Protégé berupa file OWL Memasukkan data berupa (.owl) data dari protege yang telah disimpan.
- Setelah data berhasil di *Load* maka dengan otomatis smore akan membaca rangkaian *class ontology* beserta *property* dari file tersebut.
- masukkan alamat web yang akan memuat data-data yang berkaitan dengan *individual* dari file *ontology*.
- Setelah file *ontology* dan dokumen web terkait sudah dijalankan, kemudian perlu membuat *New Individual* dan memberikan nama (ID) dari individual tersebut.
- Apabila sudah diberi penamaan maka bisa mengisi *object properties* yang didalamnya terdapat beberapa *individual* yang saling mengaitkan dengan individual yang telah dibuat data disimpan berupa metadata, selesai.

Untuk lebih jelasnya dapat dilihat pada gambar 3.10 *interface* dari proses membuat metadata menggunakan *smore*:

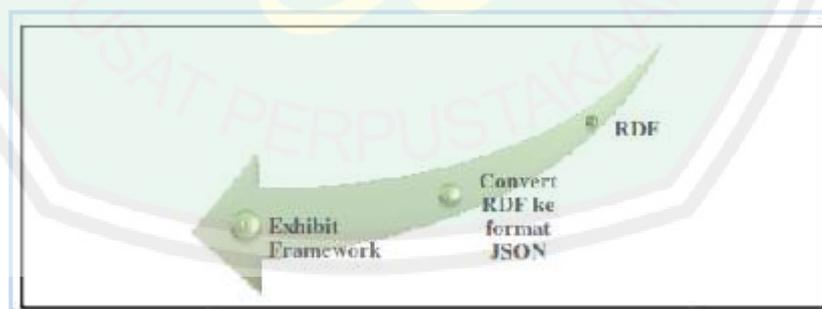


Gambar 3.10 *individual* dalam *smore*

Pembuatan metadata yang dimodifikasi dari file *ontology* ini akan membentuk dokumen yang terbagi dalam Subjek, Predikat, dan Objek yang kemudian data tersebut akan disimpan ke dalam dokumen metadata. Dokumen tersebut berupa RDF yang mana dalam bentuk pemetaannya masih berupa file OWL. Sehingga untuk penyederhanaannya perlu dibuat N-Triple dalam bentuk tabel yang akan dijelaskan pada subbab berikut.

3.5 Membuat Pemodelan dengan N-Triple

Komponen terpenting dari aplikasi *semantic web* adalah data yang akan diolah dan digunakan pada aplikasi yang dibangun. Dalam pemodelan semantik biasanya data dapat diperoleh dengan cara men-download data berupa RDF yang dipanggil melalui bantuan query SPARQL. SPARQL dikonversi menggunakan Exhibit sebagai *framework* untuk membuat aplikasi *web pages* interaktif yang didalamnya terdapat *java script*. Format asli *database* yang ada di *Exhibit* adalah JSON format seperti pada gambar 3.11 berikut:



Gambar 3.11 Proses penerapan data RDF menggunakan *exhibit framework*

(Sumber: Cahyono:2010)

Ada beberapa cara untuk menggunakan format yang berbeda. Jika mempunyai data yang sudah ada dengan format yang berbeda atau lebih memilih format yang berbeda namun, dalam penelitian ini peneliti mencoba dengan pengembangan metode lain agar pemanggilan data menjadi lebih mudah dan

3.6 Indexing, Crawling Data menggunakan Sphider

Sphider adalah *web-spider* dan mesin pencari *open-source* (<http://www.sphider.eu>). Ini mencakup *crawler* otomatis, yang dapat mengikuti *link* yang ditemukan pada sebuah situs, dan *indexer* yang membangun *indeks* dari semua istilah pencarian yang ditemukan di halaman. *Sphider* ditulis dalam PHP dan menggunakan MySQL sebagai databasenya. Sistem juga memiliki *record* tentang *keyword* yang pernah dicari oleh pengunjung. Dengan *record* ini kita juga bisa memantau kata kunci apa yang paling sering dicari.

Crawling merupakan proses bertujuan untuk menyediakan informasi konten atau keseluruhan isi halaman yang terdapat pada website <http://localhost/kampus/home.php> kedalam table di *database*. Melalui proses *crawling* sistem akan menerima data-data sebagai berikut:

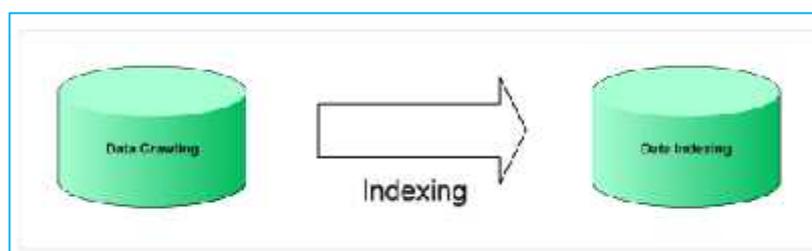
- a. *Url*: merupakan alamat url dari konten yang ter-*crawling*.
- b. *Header*: status HTTP yang diterima dari konten ter-*crawling*.
- c. *Title*: judul dari setiap konten ter-*crawling* yang diambil dari kata-kata yang diapit oleh tag `<title>... </title>`.
- d. *Content*: isi dari konten yang ter-*crawling*.

Untuk lebih jelasnya dapat dilihat pada gambar 3.13:

Sites	Categories	Index	Clean tables	Settings
Statistics	Database	Log out		
Advanced options				
Address: <input type="text" value="http://"/>				
Indexing options: <input checked="" type="radio"/> Full				
<input type="radio"/> To depth: <input type="text" value="2"/>				
<input type="checkbox"/> Reindex				
<input type="button" value="Start indexing"/>				

Gambar 3.13 *Crawling data pada Sphider*

Proses *Indexing* data merupakan proses selanjutnya setelah dilakukan *crawling* pada dokumen-dokumen web yang ada. Hasil alamat (*link*) dari proses *crawling* akan ditelusuri kembali dengan kata-kata yang ada pada *link* yang telah tersimpan dalam *database*. Penelusuran kata dilakukan berdasarkan *title* dan *content* dari setiap *link* yang ada. Banyaknya kata pada *title* dan *content* akan dihitung sesuai jumlah kata dari tiap *link*. Dari hasil penelusuran kata, selanjutnya akan disimpan dalam *database* tabel *indexing*. Dapat di ilustrasikan proses *indexing* seperti pada gambar 3.14 berikut:



Gambar 3.14 Ilustrasi proses *indexing*

pendekatan *ontology* dimana di dalam *ontology* tersebut digunakan sebagai penyimpanan dokumen dan menggunakan metode MCDM (*Multi Criteria Decision Making*) untuk menentukan bobot nilai yang akan direkomendasikan.

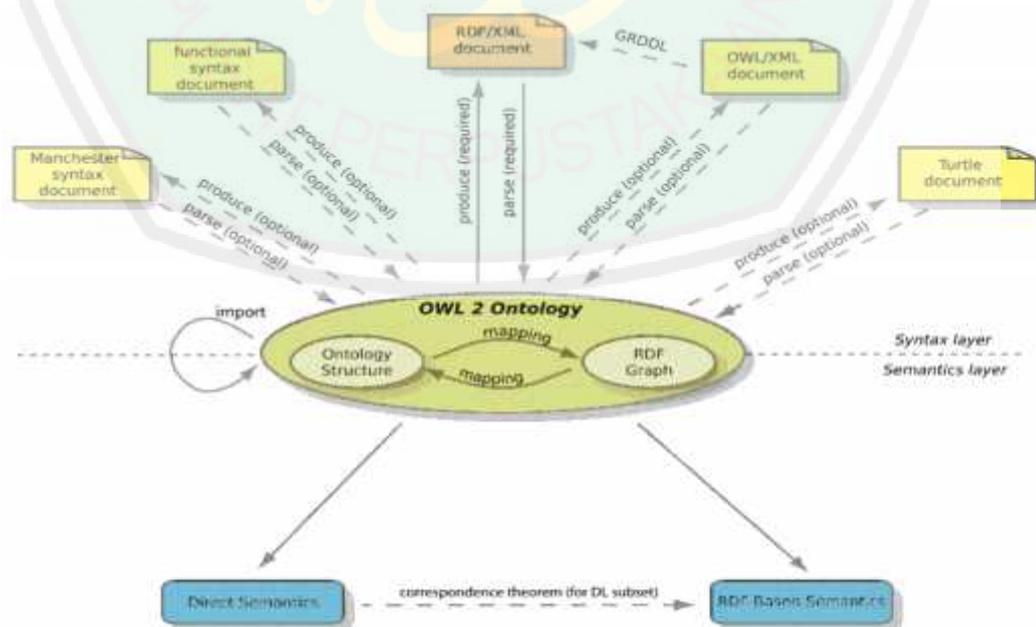
Dalam menemukan informasi yang terdapat pada *ontology*, biasanya digunakan SPARQL sebagai bahasa *query* untuk RDF/OWL namun dengan adanya improvisasi metode pemanggilan data, maka dalam penelitian kali ini proses pemanggilan menjadi lebih sederhana dengan memanggil data penyimpanan *ontology* yang dikemas dalam *table* pada database MySQL. Aplikasi akan melakukan pengecekan pada *table N-Triple* dimana di dalamnya terdapat rangkaian *ontology* yang terdiri dari kumpulan RDF yang nantinya dikonversi kedalam *table SQL* dan membentuk pola SPO (Subjek-Predikat-Objek).

Jena merupakan *framework* berbasis java yang digunakan sebagai proses pendukung konversi bentuk OWL kedalam *table N-Triple*. Sehingga apabila telah terbentuk dalam sebuah database, maka untuk menemukan informasi tersebut hanya menggunakan *query SQL* sebagai bahasa *query N-Triple*. Untuk web pagedari aplikasi ini dibangun secara manual menggunakan bahasa pemrograman PHP (*Personal Home Page*).

Penggunaan metode menggunakan teknologi domain yang dapat digunakan untuk mencari sumber informasi yang relevan dengan yang diinginkan. User yang mencari mengisi kriteria yang diinginkan. dalam server akan di cocokkan dengan domain yang ada. Di dalam nya terdapat *ontology* yang mempunyai masingdari informasi yang dicari. Secara skematis, pendekatan

penggunaan metode *semantic* untuk membangun sistem dan salah satunya menggunakan teknologi ontology akan memiliki struktur atau *hirarki* dari sebuah domain yang dapat digunakan untuk mencari sumber informasi yang relevan dengan yang diinginkan. mencari sebuah informasi, yang perlu dilakukan dengan mengisi kriteria yang diinginkan.

Web *client* akan mengirim kriteria ke *server*, di dalam *server* akan di *cocokkan* dengan domain yang ada. Di dalam nya terdapat yang mempunyai masing-masing URI yang isinya merupakan domain dari informasi yang dicari. Secara skematis, pendekatan *semantic* untuk membangun sistem dan salah satunya akan memiliki struktur atau hirarki dari sebuah domain yang dapat digunakan untuk mencari sumber informasi yang relevan sebuah informasi, yang perlu dilakukan dengan akan mengirim kriteria ke server, di dalam server akan di *cocokkan* dengan domain yang ada.



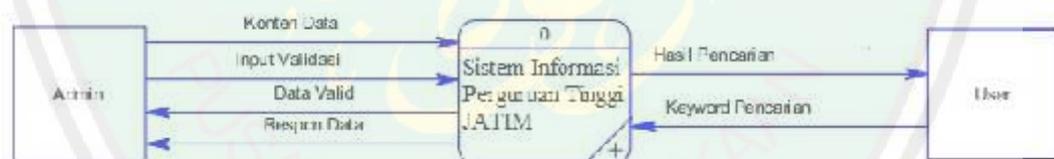
Gambar 3.15 Skema pendekatan *semantic*

(Sumber: <http://www.w3.org/TR/owl-ref/>)

3.7.2 Context Diagram

Context diagram pada aplikasi *semantic web* yang ditunjukkan pada gambar 3.16 merupakan gambaran sistem/aplikasi pada top level. Penjelasan dari *context* diagram tersebut, bahwa *user* melakukan aktivitas akses *web semantic* dan input *keyword*. Pada saat aplikasi dapat diakses, aplikasi *semantic web* akan menampilkan seluruh data Sistem Perguruan Tinggi Jatim, dimana tampilan tersebut adalah tampilan awal (*default*) dari aplikasi.

Untuk admin dapat melakukan seluruh akses data untuk dapat manajemen data pada aplikasi *semantic web*. Sebagai keamanan dalam manajemen data, maka admin perlu melakukan proses validasi pengguna, sehingga apabila akan melakukan proses pembaruan data perlu melewati proses tersebut.



Gambar 3.17 Context Diagram Aplikasi

Pada *context diagram* diatas dapat dijelaskan sistem aplikasi rekomendasi perguruan tinggi berbasis *semantic web* serta hubungan antara ketiga *external entity* yaitu: *Admin*, Sistem Perguruan tinggi JATIM, dan *User* sebagai berikut:

a. *Admin*

Pengelola sistem informasi secara keseluruhan diantaranya mendefinisikan, membuat, menghapus, mengupdate hak *administrator* secara keseluruhan.

b. Sistem Informasi Perguruan Tinggi JATIM

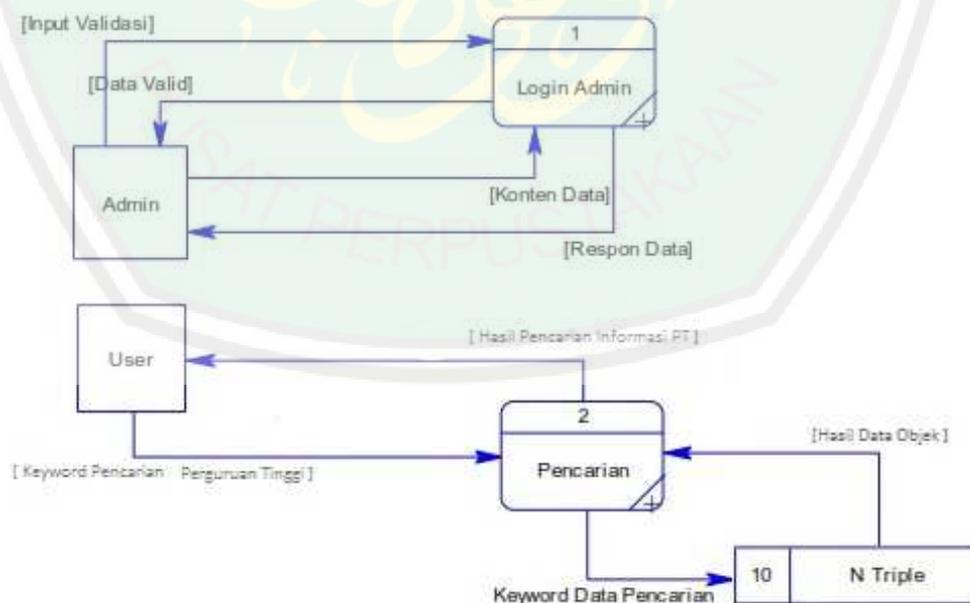
Sistem informasi akan menerima *request* (permintaan) dari admin dan membalas dengan memberi respon metadata.

c. *User/Visitor*

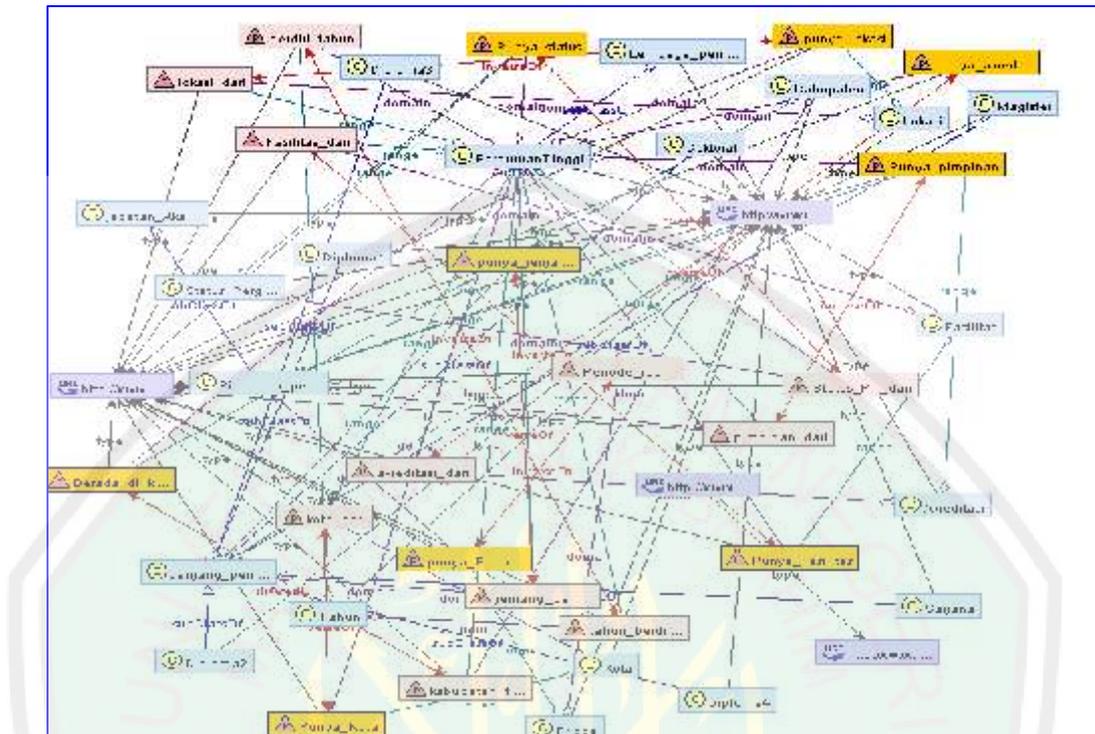
Semua pengunjung sistem, dapat mengisikan *keyword* untuk mencari informasi di sistem sesuai dengan kata kunci yang dipilih, dan menerima hasil pencarian dari sistem jika *keyword* yang dicari benar/ tepat.

3.7.3 Data Flow Diagram (DFD)

Dari *context diagram* (lihat Gambar 3.17) kemudian dijabarkan lebih dalam pada data flow diagram (DFD). Pada DFD level 1 (lihat Gambar 3.18) ditunjukkan tahapan proses yang dilakukan dalam search engine. Dan dari proses tersebut user mendapatkan informasi Perguruan Tinggi yang sesuai dengan inputan yang dimasukkan user.



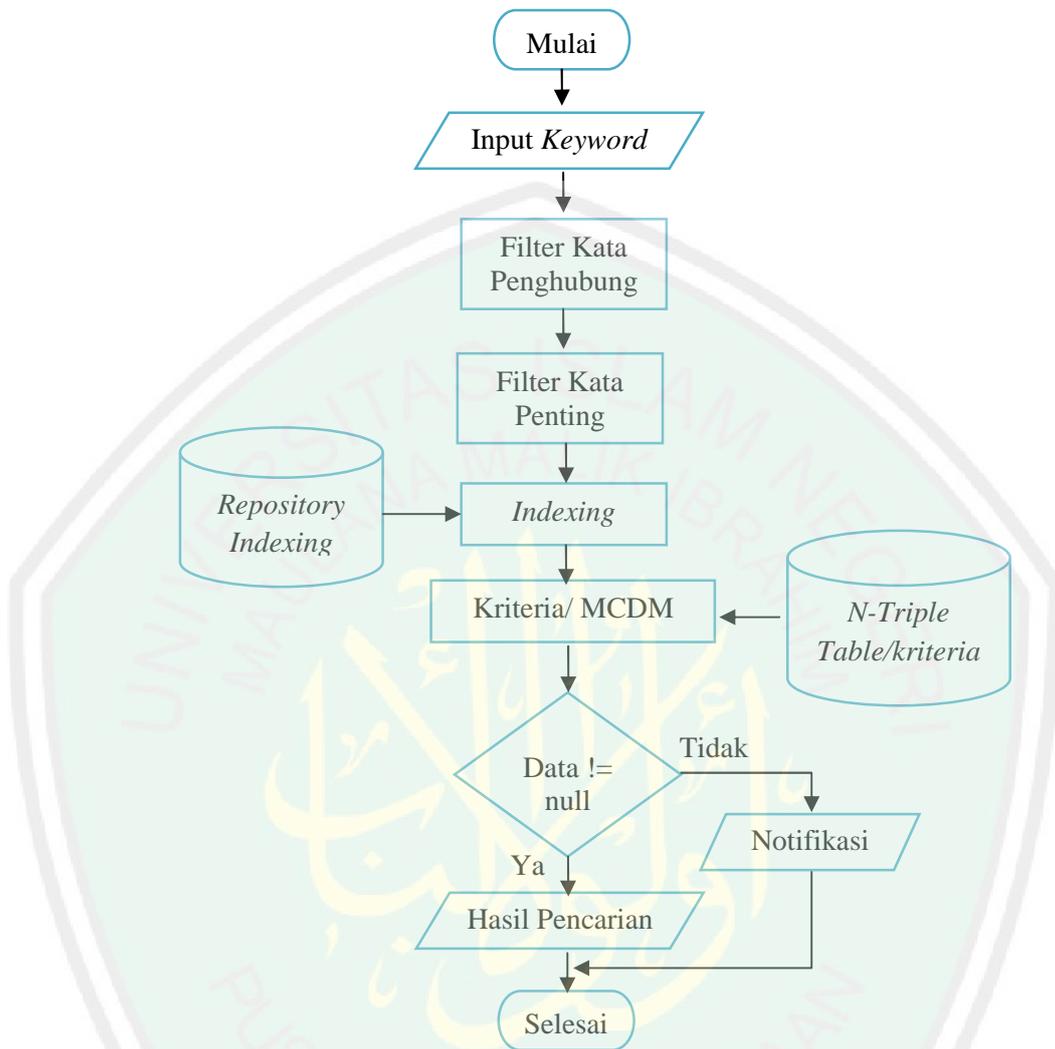
Gambar 3.18 Data Flow Diagram



Gambar 3.20 Entity relationship diagram menggunakan RDF Gravity

3.7.5 Flowchart Proses Pencarian Kriteria

Aplikasi *semantic web* bertujuan untuk menemukan informasi-informasi penting yang saling terkait di dalamnya. Proses pencarian pada sistem aplikasi rekomendasi perguruan tinggi memiliki proses yang dilakukan agar komputer dapat memahami makna yang dimaksudkan oleh pengguna. Adapaun alur dari aplikasi rekomendasi perguruan tinggi Jatim dapat melakukan pencarian dapat dijelaskan dalam beberapa langkah sesuai dengan gambar 3.21:



Gambar 3.20 Alur sistem pencarian kriteria

Metode pencarian dengan menggunakan kriteria pencarian lebih spesifik untuk mencari data dari *file* database yang telah di parsing menggunakan *N-Triple* setelah user menentukan ingin mencari informasi Rekomendasi perguruan tinggi dengan menggunakan kriteria pencarian, maka *flowchart* kriteria pencarian ini berfungsi untuk menggambarkan proses yang terjadi dalam kriteria pencarian.

3.8 Tahap Implementasi Metode MCDM dengan WPM

Perhitungan nilai preferensi untuk alternatif A_i diawali dengan memberikan nilai rating rekomendasi perguruan tinggi terhadap subkriteria ke- j (x_{ij}). Setelah masing-masing umkm diberi nilai rating kinerja, nilai ini akan dipangkatkan dengan nilai relatif bobot awal yang telah dihitung sebelumnya (w_j) dimana w_j akan bernilai positif untuk atribut *benefit* (keuntungan) dan bernilai negatif untuk atribut *cost* (biaya). Penjumlahan nilai w_j untuk setiap *subkriteria* pada kriteria yang sama akan bernilai 1 ($\sum w_j = 1$). Setelah didapat nilai preferensi untuk alternatif A_i , selanjutnya dilakukan perhitungan nilai preferensi relatif dari setiap alternatif (vektor V).

Contoh Kasus

Suatu user di provinsi Jawa Timur ingin memilih perguruan tinggi terbaik memiliki fasilitas dan kriteria terbaik dari yang akan dipilih. Ada 3 alternatif dari kriteria yang akan dipilih akan menjadi alternatif, yaitu : $A_1 = \text{Akreditasi}_A$, $A_2 = \text{Fasilitas Asrama Mahasiswa}$, $A_3 = \text{Kota Surabaya}$.

Ada 5 kriteria yang dijadikan acuan dalam pengambilan keputusan, yaitu :

$C_1 = \text{jarak dengan rumah user/kost-kostan dekat (km)}$

$C_2 = \text{kepadatan penduduk di sekitar lokasi (orang/km}^2\text{)}$

$C_3 = \text{jarak dari rumah peminat (km)}$

$C_4 = \text{jarak dengan perguruan tinggi terdekat (km)}$

$C_5 = \text{harga biaya kuliah dalam setiap semester}$

Rating kecocokan setiap alternatif pada setiap kriteria, dinilai dengan 1 sampai 5, dapat dilihat pada tabel 3.11 :

Tabel 3.11: alternatif pencocokan kriteria

No.	Alternatif/kriteria	Rating kecocokan
1.	Kriteria 1	Sangat buruk
2.	Kriteria 2	Buruk
3.	Kriteria 3	Cukup
4.	Kriteria 4	Baik
5.	Kriteria 5	Sangat baik

Menentukan parameter inputan yang tepat untuk pendukung keputusan dalam proses pemilihan perguruan tinggi, dapat ditentukan dengan menentukan nilai kriteria yang sesuai dengan alternatif pilihan di atas. Untuk lebih jelasnya dapat dilihat tabel 3.12 menunjukkan rating kecocokan dari setiap alternatif pada setiap kriteria berikut:

Tabel 3.12. Rating kecocokan dari setiap alternatif pada setiap kriteria

Alternatif	Kriteria				
	C ₁	C ₂	C ₃	C ₄	C ₅
A ₁	0,75	2000	18	50	500
A ₂	0,50	1500	20	40	450
A ₃	0,90	2050	35	35	800
Sifat	Benefit	Benefit	Benefit	Benefit	Benefit

Dari tabel di atas dapat ditentukan tingkat kepentingan setiap kriteria, juga dinilai dengan 1 sampai 5, yaitu : 1 : sangat rendah, 2 : rendah, 3 : cukup, 4 : tinggi, dan 5 : sangat tinggi

Pengambil keputusan memberikan bobot preferensi sebagai: $w = (5, 3, 4,$

4, 2) Sifat setiap kriteria adalah:

- a. Kriteria C2 (kepadatan penduduk di sekitar lokasi) dan C4 (jarak dengan perguruan tinggi) adalah kriteria keuntungan;
- b. Kriteria C1 (jarak dengan pasar terdekat), $C_3 =$ (jarak dari rumah peminat) (km), $C_5 =$ harga biaya kuliah dalam setiap semester adalah kriteria biaya.

Langkah pertama yang dilakukan adalah menghitung nilai perbaikan bobot terlebih dahulu dengan rumus sebagai berikut:

$$W_j = \frac{w_j}{\sum w_j}$$

Diketahui $w = (5, 3, 4, 4, 2)$ maka nilai perbaikan bobot nya adalah :

$$w_1 = 5/(5+3+4+4+2) = 0,28$$

$$w_2 = 3/(5+3+4+4+2) = 0,17$$

$$w_3 = 4/(5+3+4+4+2) = 0,22$$

$$w_4 = 4/(5+3+4+4+2) = 0,22$$

$$w_5 = 2/(5+3+4+4+2) = 0,11$$

Langkah kedua adalah menghitung nilai vektor S menggunakan rumus berikut:

$$S_i = \prod_{j=1}^n (x_{ij})^{w_j} \text{ dengan } i=1,2,3,\dots,m$$

$$S_1 = (0,75^{0,28}) (2000^{0,17}) (18^{0,22}) (50^{0,22}) (500^{0,11}) = 2,4187$$

$$S_2 = (0,5^{0,28}) (1500^{0,17}) (20^{0,22}) (40^{0,22}) (450^{0,11}) = 2,4270$$

$$S_3 = (0,9^{0,28}) (2050^{0,17}) (35^{0,22}) (35^{0,22}) (800^{0,11}) = 1,7462$$

Setelah vektor S dihitung, langkah selanjutnya melakukan perhitungan nilai vektor V menggunakan rumus 5. Nilai vektor V yang akan digunakan untuk perankingan dapat dihitung dengan rumus:

$$V_i = \frac{\prod_{j=1}^n (x_{ij})^{W_j}}{\prod_{j=1}^n (x_{ij}^*)^{W_j}}$$

dengan $i= 1,2,3,\dots,m$

$$V_1 = 2,4187 / (2,4187 + 2,4270 + 1,7462) = 0,3669$$

$$V_2 = 2,4270 / (2,4187 + 2,4270 + 1,7462) = 0,3682$$

$$V_3 = 1,7462 / (2,4187 + 2,4270 + 1,7462) = 0,2649$$

Nilai terbesar ada pada V_2 sehingga alternatif A_2 adalah alternatif yang terpilih sebagai alternatif terbaik. Dengan kata lain, Perguruan tinggi yang memiliki asrama mahasiswa terpilih sebagai kriteria terbaik dalam memilih perguruan tinggi.

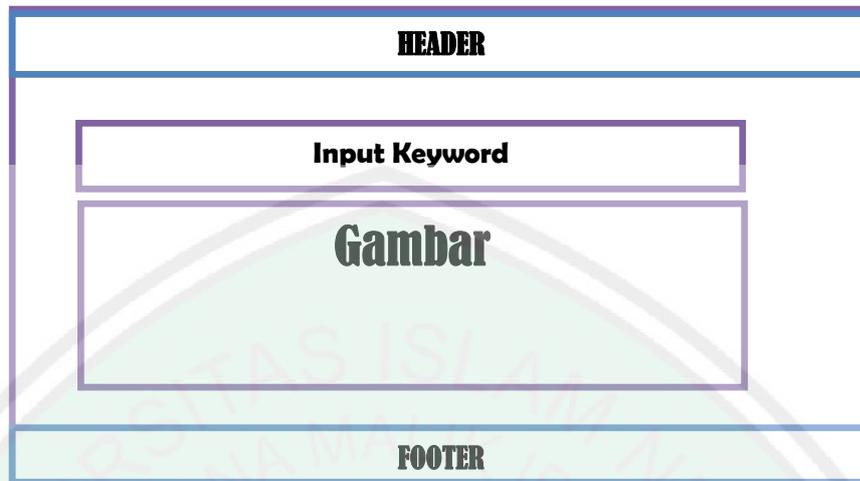
3.9 Desain Interface / Form

Desain *interface* pada aplikasi ini memiliki beberapa bagian *interface* untuk *user* dan *admin*, mulai dari halaman awal, halaman hasil pencarian, serta halaman admin dalam manajemen data.

3.9.1 Desain Interface Halaman Awal

Pada halaman awal akan menampilkan *interface* awal yang apabila aplikasi telah berhasil diakses. Di halaman sistem rekomendasi pengunjung dapat melihat kumpulan *repository* yang telah dikelola oleh administrator.

Untuk desain *interface*-nya dapat dilihat pada gambar 3.22



Gambar 3.22 Desain *Interface* Halaman pertama

Keterangan desain interface dari sistem:

a. Header

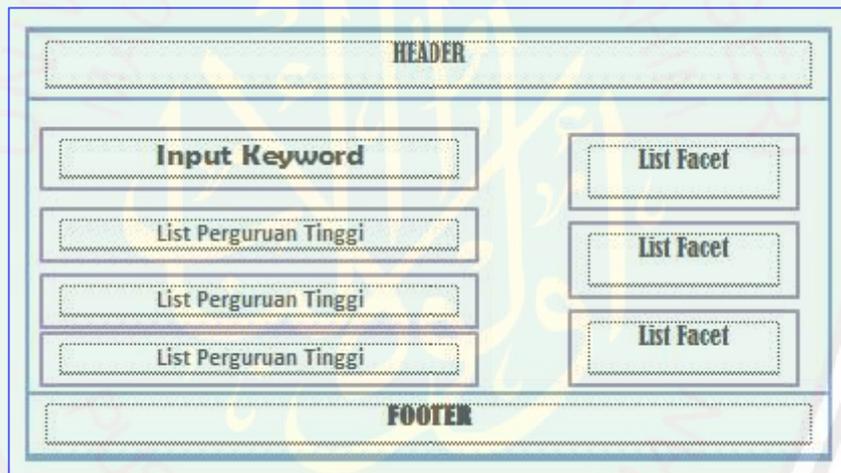
Header adalah bagian web yang paling awal dilihat pengguna. Kegunaan *header* adalah memberitahukan kepada pengguna web apa yang sedang mereka kunjungi yang berisi dengan judul dari sistem informasi.

b. Halaman ini menyajikan inputan *keyword* dari perguruan tinggi yang akan dicari (*Searching*) merupakan salah satu fungsi yang dapat memudahkan pengguna dalam menemukan informasi tertentu. Kemudahan yang diberikan oleh fungsi ini adalah efisiensi waktu, karena pengguna cukup memasukkan kata kunci yang ingin dicari dan *website* akan menampilkan halaman hasil sesuai kata kunci.

c. *footer* biasanya untuk menampilkan nomor halaman, judul dokumen, penulis dokumen, dan lain sebagainya.

3.9.2 Desain Interface Halaman Utama

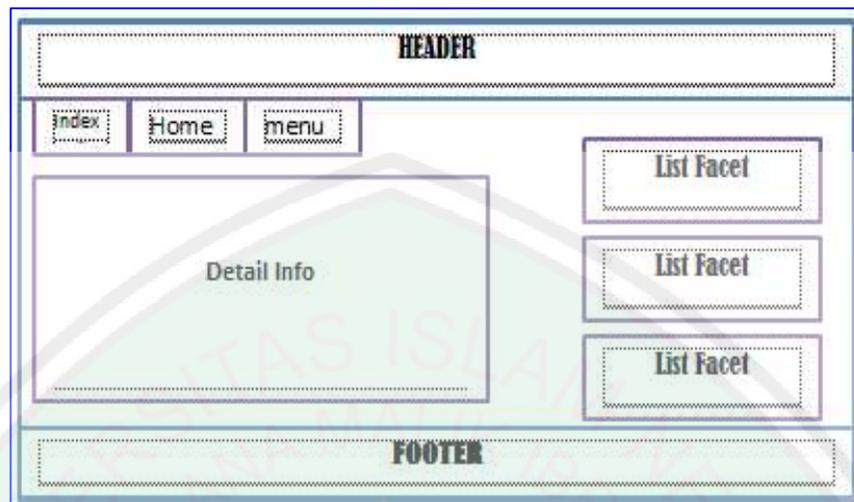
Interface pada tampilan *default*, *list* data yang muncul adalah data dari semua objek yang tersedia. Apabila pengguna meng-*input*-kan *keyword*, maka *list* data yang muncul adalah sesuai dengan *keyword* tersebut. *List* data juga dapat menampilkan data perguruan tinggi yang berkaitan dengan *facet* tertentu, misalnya *facet* “Kabupaten Malang”, maka *list* data yang muncul adalah semua data perguruan tinggi yang terdapat di Kabupaten Malang. Untuk lebih jelasnya dapat dilihat pada gambar 3.23 berikut:



Gambar 3.23 Desain *Interface* Halaman Utama

3.9.3 Desain *Interface* Halaman Detail

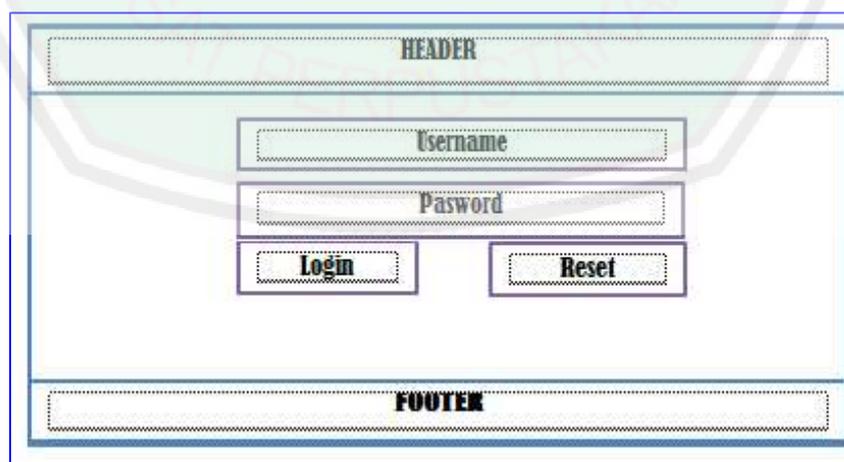
Halaman ini akan menampilkan detail info dari Perguruan Tinggi yang dipilih. Dalam halaman detail terdapat gambar beserta detail info berupa deskripsi akreditasi, jenjang pendidikan, status, tempat, alamat lokasi dan fasilitas perguruan tinggi. Untuk menampilkan rangkaian struktural terkait dengan detail objek tersebut seperti *interface* gambar 3.24 :



Gambar 3.24 Desain *Interface* Halaman Detail

3.9.4 Desain *Interface* Login Admin

Halaman Login Admin merupakan halaman untuk validasi hak akses sebagai admin. Untuk proses validasi, perlu menginputkan *username* dan *password* sebagai admin. Jika *username* dan *password* tidak valid maka admin akan diminta login kembali. Untuk lebih jelasnya bisa dilihat desain *interface panel administrator* pada gambar 3.25 berikut:



Gambar 3.25 Desain *Interface* Halaman Login Administrator

3.9.5 Desain *Interface* Edit Data

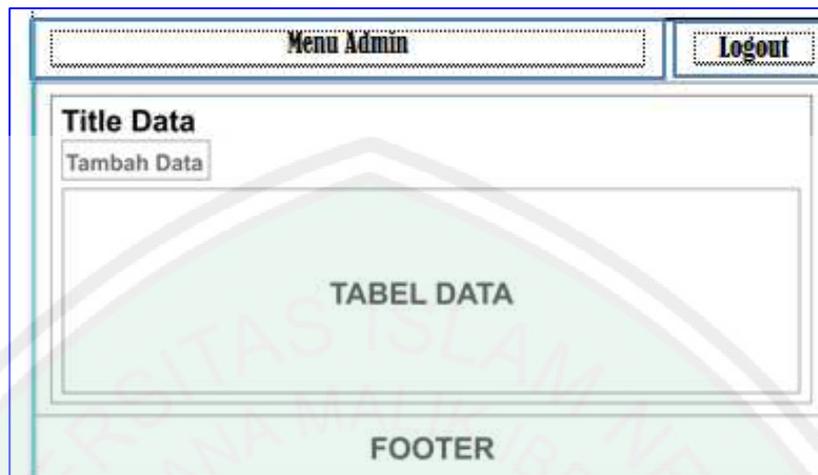
Pada halaman ini, administrator bisa mengelola data konten. Untuk mengedit data akan dilakukan di bagian edit data seperti yang terdapat di *interface* halaman edit data. Data yang dirubah adalah data yang telah dipilih sebelumnya pada *list* data yang tersedia. Sehingga pada *textfield* yang ada akan menampilkan data objek yang ingin dirubah pada data yang ada. Untuk lebih jelasnya dapat dilihat *interface* desain berikut pada gambar 3.26:

Menu Admin		Logout
Edit Title		
Edit 1	<input type="text" value="Edit Data 1"/>	
Edit 2	<input type="text" value="Edit Data 2"/>	
Edit 3	<input type="text" value="Edit Data 3"/>	
Edit ...	<input type="text" value="Edit Data: ..."/>	
<input type="button" value="Edit"/> <input type="button" value="Batal"/>		
FOOTER		

Gambar 3.26 Desain *Interface* Halaman Edit Data

3.9.6 Desain *Interface* Halaman *List* Data

Halaman *interface* yang menampilkan *list* data-data terkait dengan kebutuhan aplikasi. *List* akan ditampilkan berupa *table* yang dilengkapi dengan kolom perlakuan untuk menjalankan fungsi edit dan hapus data. Untuk menambahkan data terkait, maka dilengkapi tombol tambah data yang terdapat pada atas tabel *list* data. Data yang ditampilkan terdapat pada menu admin, *list* data yang tersedia berupa, *list* data perguruan tinggi, *list* data jenjang pendidikan dan lain-lain. Berikut ini gambar 3.27 adalah *interface* desain dari *list* data:



Gambar 3.27 Desain *Interface* Halaman List Data

3.9.7 Desain *Interface* Input Data

Informasi merupakan suatu hal yang mengalami perubahan yang sangat cepat. Seiring meningkatnya jumlah informasi pada data perguruan tinggi, mungkin perlu menata ulang atau mendistribusikan ulang informasi yang telah ada. Untuk menginputkan data, terdapat *textfield* dan komponen inputan lainnya yang digunakan untuk mengisi data dari objek yang akan di inputkan. Untuk lebih jelasnya dapat dilihat pada gambar 3.28 berikut:



Gambar 3.28 Desain *Interface* Halaman Input Data

Pada form diatas terdapat tiga fungsi tombol masing-masing tombol mempunyai fungsi sebagai berikut:

- a. Tombol “Tambah Title ” digunakan untuk menambah nilai parameter mahasiswa dan menyimpan ke database.
- b. Tombol “Batal” digunakan untuk membatalkan *input* data
- c. Tombol “Simpan” digunakan untuk menyimpan data yang telah benar dan tepat di *inputkan* kedalam database.

3.10 Perancangan Ujicoba Sistem

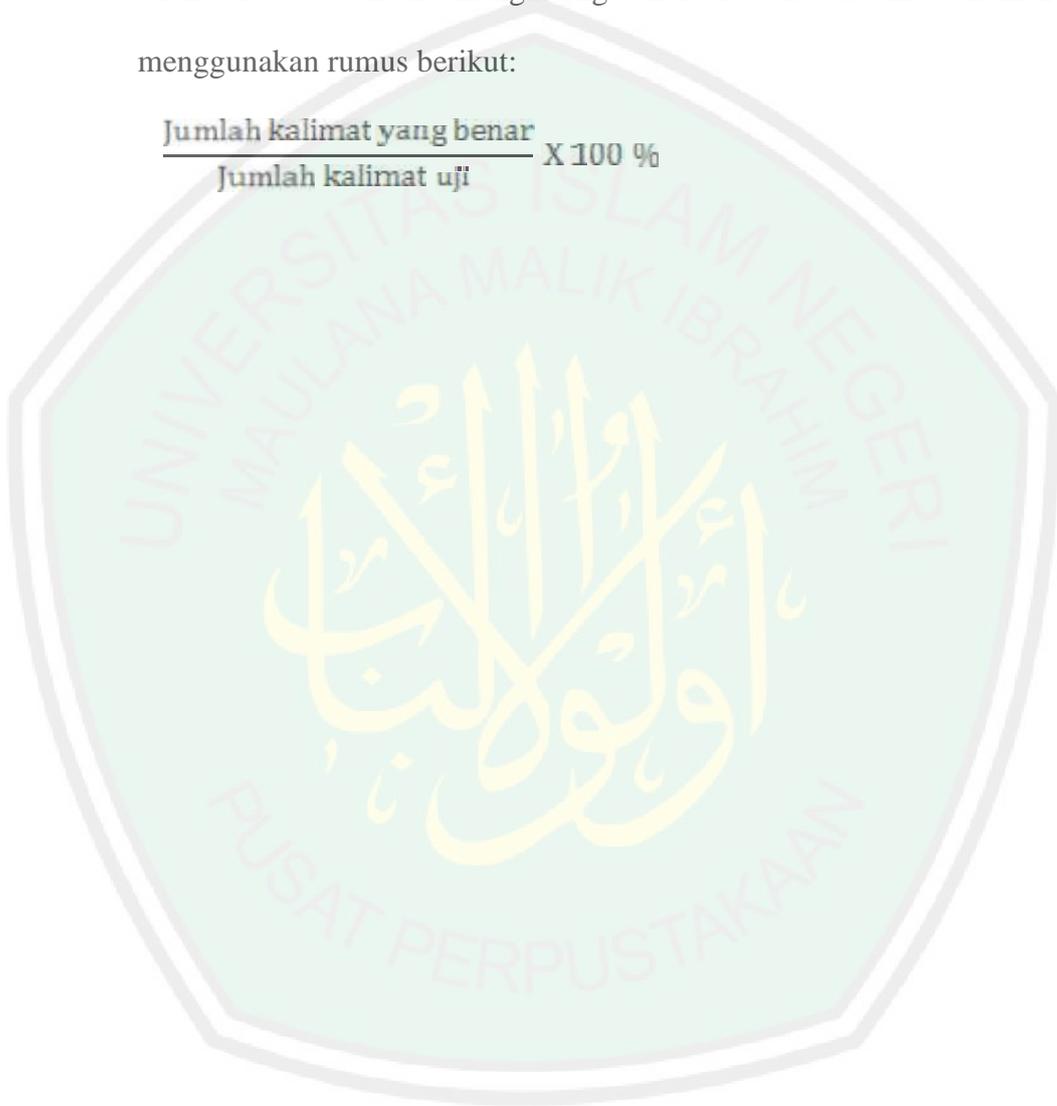
Pada subbab ini dilakukan perancangan uji coba terhadap aplikasi rekomendasi yang dibangun. Uji coba dilakukan berdasarkan input dokumen dan perhitungan nilai akurasi bobot hasil pencarian. Uji coba dilakukan berdasarkan inputan kata-kata yang membentuk kalimat dan nilai akurasi dari hasil pencarian. Tujuan dengan adanya perancangan uji coba adalah untuk mengetahui tingkat keakurasian pencarian dari sistem yang dibangun menggunakan teknologi *semantic web*.

Data yang digunakan dalam pengujian adalah sampel data sebanyak 50 data objek yang telah di inputkan sesuai dengan data Perguruan Tinggi Provinsi Jawa Timur. Dari hasil pencarian yang diperoleh aplikasi tersebut dapat dihitung nilai akurasinya. Adapun proses ujicoba yang dilakukan melalui beberapa tahapan berikut:

- a. Mengumpulkan 25 *keyword* (kata kunci) yang akan diujicobakan
- b. Menyiapkan peralatan-peralatan yang digunakan
- c. Input dokumen dan melihat hasil yang diperoleh dari mesin pencari

- d. Mencatat hasil pencarian dari sistem aplikasi
- e. Mengukur tingkat akurasi aplikasi *semantic web* berdasarkan hasil yang telah dicatat. Untuk menghitung akurasi dari sistem rekomendasi menggunakan rumus berikut:

$$\frac{\text{Jumlah kalimat yang benar}}{\text{Jumlah kalimat uji}} \times 100 \%$$



BAB IV

IMPLEMENTASI DAN PENGUJIAN

Proses implementasi dari perancangan aplikasi yang dilakukan pada bab sebelumnya akan dijelaskan pada bab ini. Implementasi ini merupakan tahapan lanjutan dari tahap perancangan yang sudah dilakukan. Dalam tahap implementasi ini akan dijelaskan mengenai tahapan-tahapan yang digunakan dalam membangun sistem ini, file-file yang digunakan dalam membangun sistem, tampilan web beserta potongan-potongan script program untuk implementasi sistem, pengujian sistem, serta evaluasi dan analisa. Berikut penjelasan mengenai tahap-tahapannya.

4.1 Implementasi Sistem

Untuk membangun sebuah aplikasi, tentunya akan mempunyai daftar kebutuhan dalam membangun sistem aplikasi tersebut. Pada tahapan ini dibahas kebutuhan-kebutuhan yang diperlukan untuk membangun aplikasi maupun untuk menjalankan aplikasi *semantic web*. Kebutuhan tersebut baik berupa perangkat lunak (*software*) maupun perangkat keras (*hardware*). Berikut ini penjelasan secara *detail*.

4.1.1 Kebutuhan Perangkat Keras (*Hardware*)

Kebutuhan perangkat keras (*Hardware*) yang digunakan untuk mendukung proses pengembangan sistem aplikasi *web semantic* ini menggunakan laptop dengan spesifikasi sebagai berikut:

- a) Laptop Axioo Neon 14"
- b) Processor Intel Centrino core 2 duo

- c) RAM (*Random Acces Memory*) 2 GB.
- d) VGA atiradeon
- e) Hardisk 250 GB

4.1.2 Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan untuk mendukung dalam pembuatan atau pengoperasian aplikasi *semantic web* ini, antara lain:

- a. Operating system, menggunakan Windows 7 Ultimate 32-bit (6.1, Build 7600)
- b. JDK (Java Development Kit) 1.6.0
- c. Protégé 3.4.5, sebagai pemodelan ontology
- d. Notepad++, 6.2.Installer sebagai script editor
- e. RDF-Gravity (version 1.0), sebagai RDF Graph
- f. Jena Framework, sebagai *library* pembentukan N-Triple
- g. Smore, sebagai *tool* dalam membentuk metadata
- h. Sphider versi 1.3.5, sebagai *tool* untuk proses *crawling*
- i. XAMPP 1.7.7, sebagai *database* penyimpanan data dan N-Triple
- j. Mozilla Firefox, *browser* yang digunakan

4.2 Pengujian Aplikasi

Pada pengujian aplikasi akan dijelaskan mengenai komponen-komponen dari aplikasi Rekomendasi Perguruan Tinggi Provinsi Jawa Timur yang diimplementasikan pada halaman *semantic web* disertai dengan fungsi dan kegunaannya. Berikut beberapa tampilan web yang terdapat dalam aplikasi.

4.2.1 Pengujian Menu Awal Sistem

Menu awal merupakan tampilan awal saat pertama kali aplikasi dijalankan. Tampilan halaman web merupakan halaman yang akan diakses langsung oleh pengunjung, seperti pada gambar 4.1 berikut:



Gambar 4.1 Tampilan Awal sistem

Berikut untuk penjelasan dari masing-masing komponen tampilan awal pada gambar 4.1 diantaranya:

a. Header

Header berfungsi sebagai penegas terkait aplikasi rekomendasi perguruan tinggi Jawa Timur. Disamping itu header juga digunakan sebagai *web cover* agar tampilan *web* menjadi lebih menarik.

b. Searchbox

Searchbox digunakan untuk memasukkan kalimat untuk mencari koleksi perguruan tinggi yang terdapat pada sistem. Pengguna dapat memasukkan kalimat hanya dengan mengetik kata-kata yang kemudian membentuk sebuah kalimat pencarian.

c. Footer

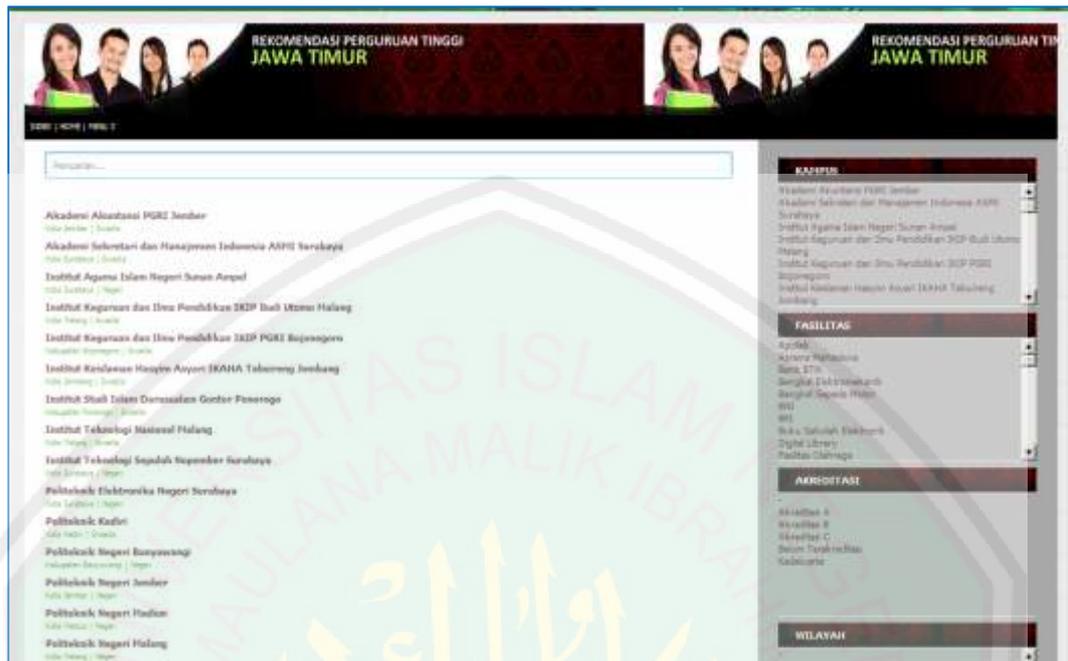
Bagian footer memberikan informasi terkait keaslian dan identitas peneliti (hak cipta) dan tahun dibuatnya aplikasi Rekomendasi perguruan tinggi ini.

Potongan *source code* untuk tampilan awal dari halaman web perguruan tinggi dapat dilihat sebagai berikut:

```
</div>
<div id="menu">
  <a href="index.php">INDEX</a> | <a href="home.php">HOME</a> | <a href="#">MENU 3</a>
</div>
<div id="content">
  <div class="kiri">
    <form name='search' method='post' action='home.php'>
      <center>
        <input class="keyword2" id="kata" name="kata" type="text" placeholder="Pencarian..."/>
      </center>
    </form>
    <br>
    <br>
  </div>
</div>
```

4.2.2 Tampilan Halaman Utama

Setelah melakukan pencarian dengan memasukkan kalimat “Perguruan tinggi Jatim “ dari halaman awal, maka *list* data Perguruan tinggi akan muncul di halaman utama. Halaman utama merupakan halaman yang menampilkan hasil pencarian dari *keyword* kalimat yang dicari. Terdapat juga kolom *facet* untuk mencari data sesuai dengan kategori tertentu. Tampilan dari halaman utama dapat dilihat pada gambar 4.2:



Gambar 4.2 Halaman Utama sistem

Potongan *source code* dari *display* halaman utama dapat dilihat sebagai berikut:

```

$queryIndex ." ") AND (count_content != 0 OR count_title !=0)
ORDER BY count_title DESC, count_content DESC";
$indexing = mysql_query($queryIndex);
$num_indexing = mysql_num_rows($indexing);
if ($num_indexing > 0)
{
$no = 1;
while ($sin = mysql_fetch_array($indexing))
{
echo "$no $sin[link]<br>";
$no++;
}
}
} elseif ($katax == ") {
$queryAll = mysql_query("SELECT
p.nama_perguruan AS nama_perguruan,
p.alamat AS alamat,
p.nama_pimpinan AS nama_pimpinan,
p.link AS link,
AND p.id_status = s.id_status

while ($qa = mysql_fetch_array($queryAll)) {
echo "
<div>
<div class='judul_content'><a href='\$qa[link]' target='_blank'>\$qa[nama_perguruan]</a></div><br>
}
} else {
echo "<div>
Data yang Dicari Tidak Ditemukan<br>
<div class='clearfix'></div>
</div>

```

4.2.3 Tampilan Halaman Detail

Pada halaman menu utama detail menampilkan informasi terkait dengan suatu objek tertentu. Info detail berisi Nama perguruan tinggi, alamat, daerah, status, akreditasi, tahun berdiri, nama pimpinan, fasilitas perguruan tinggi serta jenjang pendidikan, dari perguruan tinggi tersebut.

Berikut tampilan dari halaman detail yang ditampilkan pada gambar 4.3



Gambar 4.3 Halaman Detail

Potongan *source code* dari halaman detail dapat dilihat sebagai berikut:

```
while ($qo = mysql_fetch_array($queryObjek)){ ?>
  <div class="judul_objek">
    <h1 class="judul_detail"><?php echo "$qo[nama_objek]"; ?></h1>
  </div>
  <div class="deskripsi_objek">
    <span><?php echo "$qo[lokasi_objek]"; ?></span> <br><br>
    <p class="par_content"><?php echo "$qo[isi]"; ?></p> <br><br>
    <b>Fasilitas yang Tersedia:</b><br>
    <?php $fasQuery = mysql_query("SELECT fo.id_objek AS id_objek,
      f.nama_fasilitas AS nama, fo.id_fasilitas AS id_fas2,
      f.id_fasilitas AS id_fas, fo.checked AS checked FROM
      objek o, fasilitas_objek fo, fasilitas f WHERE
      fo.id_objek = o.id_objek AND fo.id_fasilitas = f.id_fasilitas
      AND o.nama_seo = '$nama_url' ");
    while ($fq = mysql_fetch_array($fasQuery)){
      if ($fq['checked'] == 1){
        echo "- $fq[nama] <br>";
      }
    }
  }
}
```

4.2.4 Pengujian Halaman Administrator

a. Login

Tampilan pada halaman login admin berisi input *username* dan input *password*. Untuk tipe *username* adalah *text* dan tipe *password* adalah *password*. Kemudian dilengkapi dengan tombol 'reset' apabila pengguna ingin menghapus *text* yang ada pada kedua inputan tersebut. Sebagai hasil dari tampilan halaman login admin dapat dilihat pada gambar 4.4 berikut:

Gambar 4.4 Halaman login admin

Untuk potongan *source code* dari halaman login dapat dilihat sebagai berikut:

```
<form name="form_admin" method="post" action="aksi_login.php">
  <input class="logintext" type="text" name="username" placeholder="Username"/><br>
  <input class="logintext" type="password" name="password" placeholder="Password"/><br><br>
  <table style="border: solid 0px; margin: 0 auto 0 auto; position: relative;">
    <tr><td colspan="2" align="center">
      <input class="loginsub" type="submit" name="b_login" value="Masuk"/>
      <input class="loginres" type="reset" name="b_reset" value="Reset"/>
    </td></tr></table>
</form>
```

b. Pengujian Halaman Panel Administrator

Setelah masuk ke dalam aplikasi maka selanjutnya, menu halaman *administrator* berfungsi untuk melakukan pengolahan data yang selanjutnya bisa diolah oleh administrator untuk ditampilkan di web. Menu meliputi tambah data, hapus data, maupun *update* data. Selain itu, juga terdapat menu admin yang menghubungkan manajemen data satu dengan yang lainnya. Adapun tampilan dari halaman administrator panel dapat dilihat pada gambar 4.5 berikut:



Gambar 4.5 Halaman panel Administrator

Untuk potongan *source code* dari halaman panel administrator dapat dilihat sebagai berikut:

```
<h1 class="panel_text">Administrator Panel</h1>
<div id="box_content">
  <a href='index.php?page=obj&act=add'><img src='../images/objek.png'><br>Input Objek</a>
</div>
<div id="box_content">
  <a href='index.php?page=fas&act=add'><img src='../images/fasilitas.png'><br>Input Fasilitas</a>
</div>
<div id="box_content">
  <a href='index.php?page=peng&act=add'><img src='../images/pengelola.png'><br>Input
Pengelola</a>
</div>
<div id="box_content">
  <a href='index.php?page=pos&act=add'><img src='../images/maps.png'><br>Input Posisi</a>
</div>
<div id="box_content">
  <a href='index.php?page=da&act=add'><img src='../images/daerah.png'><br>Input Daerah</a>
</div>
<div id="box_content">
  <a href='index.php?page=ad&act=add'><img src='../images/admin.png'><br>Input Admin</a>
</div>
```

c. Halaman Menu Admin

Halaman admin berfungsi untuk mengedit (*update*), menambah (*add*) dan menghapus admin. *Interface* halaman admin dapat dilihat seperti pada gambar 4.6 sebagai berikut:



Gambar 4.6 Halaman menu admin

Tampilan dari potongan source code admin.php dapat dilihat sebagai berikut:

```

</tr>
<tr>
  <td class="t">Username</td>
  <td class="t"></td>
  <td class="t"><input type="text" size="50" name="username"></td>
</tr>
<tr>
  <td class="t">Password</td>
  <td class="t"></td>
  <td class="t"><input type="password" size="50" name="password"></td>
</tr>
<tr>
  <td colspan="3" align="left" style="">
    <input type="submit" name="btn_simpan_Admin" value="Simpan">
    <input type="button" name="btn_back_Admin" value="Batal" onclick="self.history.back()">
  </td>
</tr>
</table>
</form>
<?php

```

d. List Data Perguruan Tinggi

Halaman *list* data Perguruan tinggi berisi tabel berupa data-data dari tiap konten data yang ada pada perguruan tinggi jawa timur. Kolom pada setiap tabel berisi inputan yang digunakan sebagai aksi update data dan hapus data. Untuk menambahkan data juga terdapat tombol tambah data yang ada diatas tabel. Untuk lebih jelasnya dapat dilihat pada gambar 4.7 berikut:

No	Nama Perguruan	Alamat	Perguruan	Perbaikan
1	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
2	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
3	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
4	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
5	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
6	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
7	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
8	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
9	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>
10	Mahasiswa Perguruan Tinggi	Jl. Raya 12345	12345678901234567890	<input type="checkbox"/>

Gambar 4.7 List Data Perguruan Tinggi

Potongan *source code* dari halaman salah satu contoh *list* data yaitu *list* data

‘perguruan tinggi’ dapat dilihat sebagai berikut:

```
while ($perguruan = mysql_fetch_array($queryperguruan)) {
    // inputan fas-jen >> echo "<td>$perguruan[nama_perguruan] | <a
href=?page=per&act=add_jf&idinfo=$perguruan[id_perguruan]>input</a></td>";
    echo"
    <tr>
        <td align='center'>$nomor</td>
        <td>$perguruan[nama_perguruan]</td>
        <td>$perguruan[Alamat]</td>
        <td>$perguruan[nama_pimpinan]</td>
        <td align='center'>
            <a href=?page=per&act=edit&idinfo=$perguruan[id_perguruan]><img
src='./images/edit.png'></a>&nbsp;
            <a href=perguruan_act.php?exc=del&idinfo=$perguruan[id_perguruan]><img
src='./images/delete.png'></a>
        </td>
    </tr>
    ";
    $nomor++;
}
}
```

Adapun tampilan dari *list* data dari jenjang pendidikan yang dimiliki dari perguruan tinggi adalah sebagai berikut pada gambar 4.8:

No	Nama Jenjang	Perbaikan
1	01 Pendidikan Dasar	 
2	02 Pendidikan Menengah Pertama	 
3	03 Pendidikan Atas Menengah Pertama	 
4	04 Pendidikan Menengah	 
5	05 Pendidikan Menengah Atas	 
6	06 Pendidikan Menengah Atas	 
7	07 Pendidikan Menengah Atas	 
8	08 Pendidikan Menengah Atas	 
9	09 Pendidikan Menengah Atas	 
10	10 Pendidikan Menengah Atas	 
11	11 Pendidikan Menengah Atas	 
12	12 Pendidikan Menengah Atas	 
13	13 Pendidikan Menengah Atas	 

Gambar 4.8 List Data Jenjang pendidikan

e. *Update Data/ Add Data/Delete Data*

Pada fungsi *update data* yaitu bertanda icon warna biru sedangkan icon warna merah untuk *delete* data yang dapat di *update/delete* yaitu seluruh menu yang ada di halaman panel administrator diantaranya admin, perguruan

tinggi, jenjang pendidikan, fasilitas, daerah, tahun dan lain-lain. gambar 4.9 berikut ini adalah tampilan menu update data perguruan tinggi:

The screenshot shows a web application interface for updating university data. The interface is titled "Perguruan Tinggi" and displays a form with various input fields and checkboxes. The form is organized into sections: "Nama Program", "Alamat", "Nama Program", "Daerah", "Tahun", "Aksesibilitas", "Basis", and "Fasilitas". Each section contains several input fields and checkboxes for selection.

Gambar 4.9 interface halaman update data

4.3 Implementasi Sistem

Untuk mengimplementasikan aplikasi *semantic web*, peneliti menggunakan *ontology* dimana di dalam *ontology* tersebut digunakan sebagai penyimpanan dokumen dan Jena merupakan *framework* berbasis java yang digunakan sebagai proses pendukung konversi bentuk OWL kedalam tabel *N-Triple*. Dengan menggunakan metode MCDM (*Multi Criteria Decision Making*) untuk menentukan bobot nilai yang akan direkomendasikan.

4.3.1 Hasil Proses *Crawling*

Proses *crawling* adalah proses mengambil kumpulan halaman dari sebuah web untuk dilakukan pengindeksan sehingga mendukung kinerja mesin pencari. Tujuan dari *crawler* adalah dengan cepat dan efisien mengumpulkan banyak informasi dari halaman *web* yang berguna, berikut dengan struktur *link* yang terkoneksi dengan halaman *web* tersebut. Dengan *crawler* untuk mengumpulkan

informasi yang terdapat pada halaman *website* sehingga ketika pengguna internet memasukkan kata kunci pencarian dapat dengan cepat memberikan informasi yang relevan kepada user.

Proses *crawling* yang dilakukan oleh spider yaitu dengan mengambil dokumen-dokumen web (*link*) yang ada pada domain `http://localhost/kampus/home.php` *Link-link* dari hasil proses *crawling* tersebut akan ditampilkan dan langsung disimpan dalam *database*. Untuk hasil dari proses *crawling* dapat dilihat pada gambar 4.10.



```

Table 'spider_db.domains' doesn't exist [Back to admin]

Spidering http://localhost/kampus/home.php
1. Retrieving: http://localhost/kampus/home.php at 07:42:17.
Size of page: 46.20kb. Starting indexing at 07:42:18.
Indexed
Links found: 64. New links: 64
2. Retrieving: http://localhost/kampus/ at 07:42:18.
Size of page: 2.19kb. Starting indexing at 07:42:18.
Indexed
Links found: 0. New links: 0
3. Retrieving: http://localhost/kampus/akademi-akuntansi-pgri-jember.html at 07:42:18.
Size of page: 27.21kb. Starting indexing at 07:42:18.
Indexed
Links found: 14. New links: 0
4. Retrieving: http://localhost/kampus/akademi-sekretari-dan-manajemen-indonesia-asmi-surabaya.html at 07:42:18.
Size of page: 27.05kb. Starting indexing at 07:42:18.
Indexed
Links found: 14. New links: 0
5. Retrieving: http://localhost/kampus/home.php at 07:42:18.
already in database
6. Retrieving: http://localhost/kampus/home.php?katas... at 07:42:18.
  
```

Gambar 4.10 Hasil proses *Crawling*

4.3.2 Proses *Indexing*

Indexing atau pengindeksan merupakan proses membangun basis data indeks dari koleksi dokumen. *Indexing* dilakukan terhadap dokumen sebelum pencarian dilakukan. Untuk saat ini, sistem pengindeksan secara manual mulai digantikan oleh sistem pengindeksan otomatis. *Indexing* melakukan proses pengumpulan kata-kata atau kalimat pada suatu halaman web yang telah ter-

crawling sebelumnya. Setelah proses *crawling* selesai, hasil *link-link* yang diperoleh dari domain <http://localhost/kampus/home.php> kemudian dilanjutkan pada proses *indexing*. Kata yang diperoleh dari hasil *crawling* selanjutnya di *indexing* pada artikel-artikel dan kata-kata yang terdapat pada tiap *link* terkait. Setiap kata yang sama akan dihitung banyaknya berdasarkan *title* dan *content*. Untuk tabel *indexing* dapat dilihat pada gambar 4.11 berikut:

	word	link	count_title	count_content
1950		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1954		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1957		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1958		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1959		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1960		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1961		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1962		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1963		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1964		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1965		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1966		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1968		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1969		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1979		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1980		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1981		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0
1982		http://localhost/kampus/akademi-akuntansi-pgri-jem...	0	0

Gambar 4.11 Tabel hasil *Indexing*

Untuk *source code* dalam proses *indexing*, terdapat fungsi pada „`simple_html_dom.php`“ (<http://sourceforge.net/projects/simplehtmldom/>) untuk menyeleksi dan menemukan sintaks tertentu pada *tag* yang terdapat dalam halaman web yang dibaca. Dengan memanggil *link* „`mysql_query("SELECT link FROM temp ORDER BY link ASC");`“ kemudian akan dilakukan pengecekan kata pada *tag* yang dibaca, yaitu pada \$judul, \$article, dan \$side.

Untuk lebih jelasnya dapat dilihat pada potongan *source code* berikut:

```
require 'simple_html_dom_lama.php';

$query_data = mysql_query("SELECT link FROM temp ORDER BY link ASC");
while ($data = mysql_fetch_array($query_data)) {
    $isi = $data['link'];
    $html = file_get_html($isi);

    foreach ($html->find('body') as $judul) {
        $title = $judul->find('div.judul_objek', 0)->plaintext;
        echo '<br>';
    }
    foreach ($html->find('body') as $artikel) {
        $text = $artikel->find('div.deskripsi_objek', 0)->plaintext;
        echo '<br>';
    }
    foreach ($html->find('body') as $side) {
        $sidebar = $side->find('div.isi_sidebar', 0)->plaintext;
        echo '<br>';
    }
    ...
    $count_title = count(explode($word, $title_kecil)) - 1;
    $count_body = count(explode($word, $text_kecil)) - 1;
    $count_side = count(explode($word, $sidebar_kecil)) - 1;
    $count_content = $count_body + $count_side;
    ...
    mysql_query("INSERT INTO indexing(word, link, count_title, count_content) VALUES ('$word', '$isi',
'$count_title', '$count_content')");
}
}
```

4.3.3 Pembuatan *Table N-Triple* kriteria

Untuk pembuatan N-Triple *table* dilakukan setelah pembuatan metadata membentuk pola SPO (*Subject Predicate Object*). Dalam pembuatan n-triple ini, dilakukan pada pemrograman java dengan bantuan API Jena. Jena merupakan *library* yang dapat memfasilitasi dalam melakukan konversi dari metadata bentuk OWL menjadi *database SQL*. Untuk memisahkan dokumen dalam bentuk subject, predikat, objek menggunakan sintaks StringTokenizer dengan string pemisah yang dibutuhkan, contohnya StringTokenizer st2 = new StringTokenizer(data2, "< >");

Berikut potongan *source code* untuk memisah metadata agar membentuk kolom Subjek Predikat dan Obejk dalam tabel N-Triple.

```
StringTokenizer st = new StringTokenizer(data, "[ ]");
st.nextToken();
while (st.hasMoreTokens()) {
    String data2 = st.nextToken();
    StringTokenizer st2 = new StringTokenizer(data2, " , <>");
    while (st2.hasMoreTokens()) {
        String S = st2.nextToken();
        String P = st2.nextToken();
        String O = st2.nextToken();
        String link = "";
        String second = "";
        StringTokenizer st3 = new StringTokenizer(S, "#");
        String first = st3.nextToken();

        boolean cek = st3.hasMoreTokens();
        if (cek == true) {
            second = st3.nextToken();
            link = jTextField1.getText()+"/" + second + ".html";
        }
        System.out.println("Link : " + link);
        System.out.println("Subject : " + S + " Predicate : " + P + " Object : " + O);
        String sql = "INSERT INTO ntriple VALUES('" + S + "','" + P + "','" + O + "','" + link + "')";
        statement.executeUpdate(sql);
        statement = (Statement) conn.createStatement();
        System.out.println(" data berhasil disimpan");
        link = "";
    }
}
```

4.3.4 Filter Kata Utama

Kata penting yang ada merupakan data yang diperoleh berdasarkan hasil *crawling* yang dilakukan pada tahapan sebelumnya. *Web crawler* dimulai dengan sekumpulan URL, kemudian mendownload setiap halamannya, mendapatkan link dari setiap halaman yang dikunjungi kemudian mengulangi kembali proses *crawling* pada setiap *link* halaman tersebut. Proses *crawling* berlangsung terus sampai antrian URL kosong atau kalau kondisi berhenti sudah terpenuhi.

Untuk menyesuaikan dengan kebutuhan, perlu sedikit modifikasi agar data yang tersimpan merupakan data yang dapat digunakan dalam proses penyaringan kata penting. Sintaks yang digunakan sama seperti fungsi pada penyaringan kata penghubung, namun nilai *boolean* yang diambil adalah yang bernilai *false*.

Berikut potongan *source code* yang digunakan untuk melakukan penyaringan kata penting.

```
StringTokenizer st = new StringTokenizer(data, "[ ]");
st.nextToken();
while (st.hasMoreTokens()) {
    String data2 = st.nextToken();
    StringTokenizer st2 = new StringTokenizer(data2, ", <>");
    while (st2.hasMoreTokens()) {
        String S = st2.nextToken();
        String P = st2.nextToken();
        String O = st2.nextToken();
        String link = "";
        String second = "";
        StringTokenizer st3 = new StringTokenizer(S, "#");
        String first = st3.nextToken();

        boolean cek = st3.hasMoreTokens();
        if (cek == true) {
            second = st3.nextToken();
        }
    }
}
```

4.3.5 Pengindeksan Data

Setelah melalui proses *filtering*, kemudian akan dilakukan pengindeksan yang dicari berdasarkan jumlah *count_title* serta jumlah *count_content*. Apabila jumlah diantara keduanya bernilai '0' maka data tidak akan terseleksi dan sebaliknya bila tidak bernilai '0' maka akan dilakukan pengurutan berdasarkan jumlah kata yang cocok dalam *title* dan *content*.

Source code dari pengindeksan adalah sebagai berikut.

```
$queryIndex = "SELECT DISTINCT(link) FROM indexing WHERE ( ";
for ($a = 0; $a < $hitung_kataFilter2; $a++) {
    $queryIndex .= "(word = '$kataFilter2[$a]')";
    if ($a < $hitung_kataFilter2 - 1) {
        $queryIndex .= " OR ";
    }
}
$queryIndex .= ") AND (count_content != 0 OR count_title !=0)
ORDER BY count_title DESC, count_content DESC ";
$indexing = mysql_query($queryIndex);
$num_indexing = mysql_num_rows($indexing);
```

4.3.6 Pencocokan Kata

Pencocokan kata pada tahap ini dilakukan setelah semua proses mulai dari *filtering* kata hingga pengindeksan selesai, yaitu dengan mencocokkan kata *array* hasil proses sebelumnya dengan data pada tabel N-Triple. Untuk pencocokan kata dilakukan dengan metode yang sederhana, yaitu dengan menggunakan *query* 'LIKE' pada MySQL maka *array* yang menjadi masukan langsung dapat dicocokkan dengan data yang terdapat dalam tabel N-Triple.

Berikut potongan *source code* dari pencocokan kata.

```

for ($a = 0; $a < $hitung_kataFilter2; $a++) {
    $queryTriple = mysql_query("SELECT DISTINCT(link) FROM ntriple WHERE objek LIKE
'%$kataFilter2[$a]%'");
    while ($qt = mysql_fetch_array($queryTriple)) {
        $array_ntriple[$an] = $qt['link'];
        $an++;
    }
}
$num_ntriple = mysql_num_rows($queryTriple);
$count_ntriple = count($array_ntriple);
if ($num_ntriple > 0) {
    $tambah = 0;
    for ($c = 0; $c < $count_ntriple; $c++) {
        for ($d = 0; $d < $count_ntriple; $d++) {
            if ($array_ntriple[$c] == $array_ntriple[$d]) {
                $tambah++;
            }
            if ($tambah == $hitung_kataFilter2) {
                $array_ntriple2[$an2] = $array_ntriple[$d];
                $an2++;
            }
        }
    }
    $tambah = 0;
}

...
$stemp = "";
for ($f = 0; $f < $count_ntriple2; $f++) {
    $data = $array_ntriple2[$f];
    if ($data != $stemp) {
        $array_ntriple3[$an3] = $data;
        $an3++;
        $stemp = $data;
    }
}

```

4.4 Implementasi Antar Muka Sistem dengan MCDM

Untuk mengimplementasikan aplikasi *semantic web* dimulai dengan pencarian. Pencarian dimulai dengan memasukkan kalimat yang nantinya akan melewati tahapan *filter* kata penghubung beserta kata penting yang sesuai dengan data yang terdapat dalam *database* dan setelah itu baru dilakukan proses pencocokan kata sesuai dengan tabel N-triple dan keputusan.

4.4.1 Menentukan Alternatif Keputusan

Setiap pengambil keputusan yang ideal, akan membeberkan semua kemungkinan pilihan yang ada dan kemudian memilih satu diantaranya yang akan memberikan hasil yang terbaik bagi pencapaian tujuan. Variabel alternatif keputusan sebelumnya sudah tersaring dalam `$hitung_kataFilter2` sehingga untuk menentukan hasilnya, maka perlu beberapa kriteria keputusan yang telah tersimpan dalam tabel *indexing*. Penghitungan dilakukan secara berulang pada tiap variabel keputusan yang ada. Potongan *source code* dari *variabel* keputusan dapat di lihat pada uraian *code* berikut.

```

$hitung_kataFilter2 = count($kataFilter2);
if ($hitung_kataFilter2 != 0) {
    $queryIndex = "SELECT DISTINCT(link) FROM indexing_kampus WHERE ( ";
    for ($a = 0; $a < $hitung_kataFilter2; $a++) {
        $queryIndex .= "(word = '$kataFilter2[$a]')";
        if ($a < $hitung_kataFilter2 - 1) {
            $queryIndex .= " OR ";
        }
    }
    $queryIndex .= ") AND (count_content != 0 OR count_title !=0)
    ORDER BY count_title DESC, count_content DESC";
    $indexing = mysql_query($queryIndex);
    $num_indexing = mysql_num_rows($indexing);
    if ($num_indexing > 0) {
        $no = 1;
        while ($in = mysql_fetch_array($indexing)) {
            //echo "$no) $in[link]<br>";
            $no++;
        }
    }
}

```

4.4.2 Penentuan Nilai Bobot

Proses MCDM aplikasi rekomendasi perguruan tinggi ini dimulai dengan proses memasukkan nilai bobot tiap kriteria. Setelah proses memasukkan nilai bobot setiap kriteria disimpan, kemudian dilakukan proses penghitungan nilai intensitas akhir. Untuk pembobotan dilakukan dengan perhitungan jumlah variabel yang muncul pada setiap *criteria*, sehingga dilakukan proses pengurutan variabel sesuai dengan nilai bobot yang lebih tinggi.

Adapun potongan *source code* dari variabel keputusan dapat di lihat pada uraian *code* berikut:

```

for ($a = 0; $a < $hitung_kataFilter2; $a++) {
    $queryTriple = mysql_query("SELECT DISTINCT link FROM ntriple_kampus WHERE objek LINK '$kataFilter2[$a]'");
    while ($qt = mysql_fetch_array($queryTriple)) {
        $array_ntriple[$an] = $qt['link'];
        $an++;
    }
}
$num_ntriple = mysql_num_rows($queryTriple);
$count_ntriple = count($array_ntriple);

if ($count_ntriple > 0) {
    $tambah = 0;
    for ($c = 0; $c < $count_ntriple; $c++) {
        for ($d = 0; $d < $count_ntriple; $d++) {
            if ($array_ntriple[$c] == $array_ntriple[$d]) {
                $tambah++;
                if ($tambah == $hitung_kataFilter2) {
                    $array_ntriple2[$an2] = $array_ntriple[$d];
                    $an2++;
                }
            }
        }
    }
    $tambah = 0;
}
}

```

Agar nilai variabel yang muncul tidak bernilai ganda, maka perlu dilakukan pencocokan kembali berdasarkan *criteria* keputusan yang terdapat dalam tabel *n-triple*. Proses tersebut dilakukan secara bertahap pada tiap variabel. Apabila telah selesai pada variabel pertama, selanjutnya akan dilakukan proses pencocokan ulang sesuai *criteria* keputusan yang ada pada variabel *alternative*

selanjutnya. Potongan *source code* dari *variabel alternatif* keputusan dapat dilihat pada uraian *code* berikut:

```

$count_ntriple2 = count($array_ntriple2);
for ($e = 0; $e < $count_ntriple2; $e++) {
    sort($array_ntriple2);
}

$temp = "";
for ($f = 0; $f < $count_ntriple2; $f++) {
    $data = $array_ntriple2[$f];
    if ($data != $temp) {
        $array_ntriple3[$an3] = $data;
        $an3++;
        $temp = $data;
    }
}

$count_ntriple3 = count($array_ntriple3);
$noo = 1;
//echo "<br>Hasilnya:<br>";
for ($g = 0; $g < $count_ntriple3; $g++) {
    //echo "$noo) $array_ntriple3[$g]<br>";
    $noo++;
    $array_ntriple4[$an4] = $array_ntriple3[$g];
    $an4++;
}
}

```

Apabila dalam proses pencocokan variabel tidak memenuhi pada setiap *criteria* keputusan yang ada, maka data Perguruan Tinggi yang tidak mengandung beberapa *criteria* tersebut tidak dimunculkan. Sehingga yang dimunculkan hanyalah data yang *variabelnya* memenuhi semua *criteria* yang bersangkutan.

4.5 Uji coba Pencarian Perguruan Tinggi

Untuk mengetahui apakah sistem yang telah dibuat telah sesuai dengan kebutuhan sistem, maka dilakukan tahapan uji coba. Proses uji coba seluruh spesifikasi terstruktur dari sistem pendukung keputusan secara keseluruhan

memastikan bahwa sistem yang telah dibuat sesuai dengan rancangan sistem yang telah ditetapkan dan sesuai dengan akurasi yang maksimal.

Untuk data yang di uji terdiri dari 50 data Perguruan tinggi yang berada tersebar di Provinsi Jawa Timur. Data lengkapnya dapat dilihat pada halaman lampiran. Tahapan uji coba yang digunakan merupakan uji coba *funksional* yaitu mencari Perguruan tinggi berdasarkan kriteria tertentu seperti yang terdapat pada tabel 4.1 hasil uji coba:

Tabel 4.1: Hasil uji coba *funksional* sistem

NO	KALIMAT INPUTAN	HASIL	
		JUMLAH	KETERANGAN
1	Universitas	29	Benar
2	Institut	7	Benar
3	Akreditasi_b	16	Benar
4	Universitas dengan akreditasi_b	8	Benar
5	S1 Teknik Informatika	15	2 Salah (Bukan S1 Teknik Informatika)
6	S1 Teknik Informatika di Malang	4	Benar
7	Jurusan S1 Pendidikan Matematika	19	6 Salah (Bukan S1 Pendidikan Matematika)
8	Jurusan S1_Pendidikan_Matematika	13	Benar
9	Kampus dengan jurusan S2 Pendidikan	13	3 Salah (Bukan S2 Pendidikan)
10	Kampus yang ada jurusan S2_Pendidikan_Matematika	3	Benar
11	Universitas yang ada jurusan S1_Kimia	7	Benar
12	Institut yang akreditasi_b	4	Benar
13	Institut yang akreditasi_b di daerah Malang	2	Benar
14	Kampus dengan akreditasi_a di Malang	2	Benar
15	Kampus yang memiliki asrama	9	Benar

16	Universitas Islam yang memiliki asrama	3	Benar
17	Kampus yang memiliki asrama dan akreditasi_a	3	Benar
18	Perguruan Tinggi yang berstatus Negeri	20	Benar
19	Perguruan tinggi negeri di Jember	4	Benar
20	Perguruan tinggi Islam di Malang yang negeri	1	Benar
21	Perguruan tinggi di Surabaya yang negeri dan memiliki akreditasi_a	5	Benar
22	Perguruan tinggi Islam yang punya asrama dan jurusan S2	4	1 Salah (Bukan Perguruan tinggi Islam)
23	Kampus yang punya jurusan S2 dengan akreditasi_b dan ada BRI	2	Benar
24	Kampus Negeri dengan fasilitas kantor pos	3	Benar
25	Kampus di Surabaya yang negeri dan akreditasi_a serta ada jurusan S1_Teknik_Informatika	2	Benar

Dari hasil percobaan diatas telah dilakukan 25 kali masa percobaan dengan memasukkan kalimat yang berbeda. Hasil yang diperoleh adalah 21 kali benar dan 4 lainnya masih salah. Hasil salah merupakan data yang tidak muncul pada hasil pencarian. Hasil salah adalah data yang tidak sesuai dengan kalimat yang di inputkan dan data yang salah tidak hanya pada satu data melainkan juga lebih dari satu data. Untuk hasil kurang yaitu hasil pencarian yang sudah muncul namun hanya pada sebagian data dan data-data terkait lainnya tidak muncul.

Kesalahan yang muncul dalam aplikasi ini diakibatkan adanya beberapa kata yang memang harus memiliki kaitan satu sama lain, seperti 'akreditasi a'. Kata 'akreditasi a' apabila dilakukan pencarian maka kata akan dipisah antara 'akreditasi' dan 'a' sehingga hasil pencarian tidak belum bisa spesifik untuk mencari dari pemisahan kata tersebut. Untuk itu, dalam pencarian kata 'akreditasi a' tidak bisa menggunakan spasi dan harus diberi tanda garis bawah (_) seperti

‘akreditasi_A’. begitu juga dalam pencarian jenjang pendidikan seperti ‘S1 teknik informatika’.

Mesin pencari akan memisah kata antara ‘S1’, ‘Teknik’, dan ‘Informatika’ dan hasil pencarian yang muncul adalah perguruan tinggi yang mengandung antara tiga kata tersebut. Berbeda apabila dalam pencarian menuliskan kalimat ‘S1_Teknik_Informatika’ maka hasil pencarian yang muncul adalah perguruan yang mengandung kalimat S1 Teknik Informatika, bukan dari hasil pemecahan yang masih mengandung spasi.

Dari hasil uji coba pencarian dilakukan menginput 25 kalimat yang berbeda, Hasil yang diperoleh adalah 21 kali benar dan 4 lainnya masih salah dengan menunjukkan tingkat keakuratan tepat. Sehingga dapat diperoleh prosentase kebenaran dari aplikasi ini yaitu 84%, dengan perhitungan sebagai berikut:

Persentase pengujian kalimat yang benar:

$$\frac{\text{Jumlah kalimat yang benar}}{\text{Jumlah kalimat uji}} \times 100 \%$$

$$\text{Jadi diperoleh persentase kalimat} = \frac{21}{25} \times 100 \%$$

$$= 0,88 \times 100 \%$$

$$= 84 \%$$

Hasil tersebut masih tergolong baik dalam menemukan sebuah perguruan tinggi dengan kriteria-kriteria tertentu sehingga sangat dapat membantu pengguna untuk menentukan perguruan tinggi sesuai dengan kriteria-kriteria yang diinginkan.

4.6 Evaluasi Sistem

Sistem Rekomendasi perguruan tinggi yang telah dibuat ini masih jauh dari sempurna, hanya menghasilkan informasi perguruan tinggi berupa, nama, alamat, akreditasi, pimpinan, alamat, jenjang pendidikan serta tahun berdiri dari perguruan tinggi. Untuk pengembangan kedepannya diharapkan peneliti untuk menambah akreditasi setiap jurusan dan sesuai diharapkan, begitu juga dengan metode *Multi Criteria Decision Making* (MCDM) seperti yang diharapkan.

Sistem rekomendasi perguruan tinggi yang telah dibuat walau berusaha untuk membuat sebaik mungkin dan akurat namun tidak lepas dari kekurangan. Kemampuan berfikir manusia dalam menguasai ilmu sangatlah kecil dibandingkan dengan ilmu Allah. Kesempurnaan hanya milik Allah semata, Maha Pencipta tiada yang sanggup untuk menandingi dan tidak ada sesuatu dzat manapun yang mampu menyamainya tak terjangkau dan terbayangkan oleh akal pikiran tiada batas. Namun manusia diciptkan untuk selalu mencari tau tentang kekuasaan Allah dan mencari Ridha Allah, dalam hal ini Allah berfirman surat al-mumthanah ayat 1:

إِنْ كُنْتُمْ خَرَجْتُمْ جِهَادًا فِي سَبِيلِي وَابْتِغَاءَ مَرْضَاتِي تُسِرُّونَ إِلَيْهِم بِالْمَوَدَّةِ وَأَنَا أَعْلَمُ

بِمَا أَخْفَيْتُمْ وَمَا أَعْلَنْتُمْ وَمَنْ يَفْعَلْهُ مِنْكُمْ فَقَدْ ضَلَّ سَوَاءَ السَّبِيلِ ﴿١﴾

Artinya: ... “jika kamu benar-benar keluar untuk berjihad di jalan-Ku dan mencari keridhaan-Ku (janganlah kamu berbuat demikian). kamu memberitahukan secara rahasia (berita-berita Muhammad) kepada mereka, karena rasa kasih sayang. aku lebih mengetahui apa yang kamu sembunyikan dan apa yang kamu nyatakan. dan Barangsiapa di antara kamu yang melakukannya, Maka Sesungguhnya Dia telah tersesat dari jalan yang lurus” (QS.Al Mumtahanah:1)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Aplikasi rekomendasi perguruan tinggi ini berbasis pencarian menggunakan *semantic web* sebagai hasil dari uji coba dapat diperoleh prosentase kebenaran dari aplikasi ini yaitu 84%. Sehingga hasil pencarian yang didapatkan tepat dan akurat serta dapat menjadi lebih spesifik sesuai dengan penulisan kalimat yang dimasukkan untuk menentukan perguruan tinggi sesuai dengan kriteria-kriteria yang diinginkan.

5.2 Saran

Penelitian ini merupakan penelitian yang cukup berpotensi untuk terus dikembangkan. Beberapa saran dari peneliti untuk pengembangan selanjutnya, yaitu:

1. Dapat dibuat konsep dan relasi yang lebih kompleks sehingga dapat meningkatkan hasil pencarian pada tingkat keakurasian yang lebih baik.
2. Aplikasi rekomendasi pencarian perguruan tinggi ini didukung oleh *GIS* (*Geographical Information System*) yang dapat dikembangkan oleh peneliti selanjutnya.

DAFTAR PUSTAKA

- Ali.2006:*Tips Memilih Perguruan Tinggi:Yogyakarta*
- Apache Jena. *About Jena*.The Apache Software Foundation.
http://jena.apache.org/about_jena/about.html (Diakses 8 Oktober 2013)
- B. Parsia A. Kalyanpur, and J. Golbeck. 2005. *SMORE – Semantic Markup, Ontology, and RDF Editor*
<http://www.mindswap.org/papers/SMORE.pdf>. (Diakses pada tanggal 18 oktober 2013)
- Cahyono, Edi.2012. Implementasi Teknologi Semantik Web Untuk Membangun *Search Engine* Pada *Digital Library Slims* (Senayan *Library Management system*).diglib:uin-malang
- Fadjar, Efendi.2007. *Dunia Perguruan Tinggi dan Kemahasiswaan*:Ummpress
- Feriansyah,2011. *Perancangan sistem pendukung keputusan penentuan performance sekolah dengan menggunakan metode fuzzy multi Criteria decision making (mcdm)*. Universitas Sumatera Utara
<http://repository.usu.ac.id/bitstream/123456789/26854/3/Chapter%20II.pdf>
 (Diakses tanggal 26 Mei 2013)
- Ferdila, Mustikasari dan Ferdi. 2011. *Aplikasi Web Simantik Untuk Pencarian Materi Perkuliahan*. Universitas Gunadarma
 (Diakses tanggal 16 April 2013)
- Giese, Martin. 2011. *Semantic Days 2011 Tutorial Semantic Web Technologies Lecture 5: Presenting Relational Databases as RDF*. University Of Oslo.
- Gupta,Thakur. 2010. *Semantic Query Optimisation With Ontology Simulation*. International journal of web & semantic technology (IJWest). Bharati Vidyapeeth’s College of Engineering, New Delhi, India
 (Diakses tanggal 9 April 2013)

- Haidhir.2010. *Hadist Arba'in Nawawiyah*. Versi Indonesia:Maktab Dakwah dan Bimbingan Jaliyat Rabwah. Islamhouse.com
- Harmelen, Deborah L., W3C. 2004. *OWL Web Ontology Language Overview*. Tersedia di <http://www.w3.org/TR/2004/REC-owl-features-20040210/> (Diakses pada 15 Maret 2013)
- Indra,Hervido,Herlina.2010:*Penerapan teknologi semantic web Untuk menentukan pilihan jalur bis trans jogja:*
http://repository.upnyk.ac.id/409/1/D14_penerapan_teknologi_semantic_web_untuk_menentukan_pilihan_jalur_bis_trans_jogja.pdf (Diakses pada tanggal 11 Maret 2013)
- Ike.2007: *Peran Strategis Public Relations di Perguruan Tinggi*, Uin-Maliki Press
- Indrajit,Rusli dan Darma.2002. *Pemanfaatan Search Engine Sebagai Sarana Penunjang Proses Pembelajaran*. Andi:Jogjakarta
- Ibrahim,Niko.2007. *Pengembangan Aplikasi Semantic Web Untuk Membangun Web yang Lebih Cerdas*. Bandung
Jurnal Informatika, Vol. 3, No.1, Juni 2007:27-39
- Kalyanpur, A. dkk.2006.SMORE: *Semantic Markup, Ontology, and RDF Editor*.University of Maryland, College Park (Diakses pada tanggal 12 Oktober 2013)
- Kepres.2003. *Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional*. Lembaran Negara Republik Indonesia Tahun 2003 Nomor 78:Jakarta
- Kiran,B.asa dan Nao,Rageswara.2013. *Life Cycle Assessment (LCA) and Multi-Criteria Decision Making (MCDM) for Planning, Designing and Commissioning of Green Buildings*.
International Journal of Advanced Trends in Computer Science and Engineering. Lords Institute of Engineering and Technology, Hyderabad
<http://warse.org/pdfs/2013/icacsesp86.pdf> (Diakses tanggal 27 Mei 2013)

- Lentera Hati, Ciputat, 2002. Vol 6 (di unduh pada tanggal 16 September 2013)
- Manola, Miller.2004. *W3C Recommendation*. <http://www.w3.org/TR/rdf-primer/> (diakses pada tanggal 15 maret 2013)
- Mujia,Rahardjo.2010.*Pemikiran Kebijakan pendidikan Kontemporer*.Uinpress: Malang
- Niko, ibrahim. 2007. *Pengembangan Aplikasi Semantic Web Untuk Membangun Web yang Lebih Cerdas*. Jurnal Informatika:Bandung
- Rindiani,Tuti dan Muis,Tamsil.2010. *Bimbingan Kelompok Dengan Teknik Diskusi Kelompok Untuk Membantu Siswa Dalam Kemantapan Pengambilan Keputusan Studi Lanjut*.Unesa
http://ppb.jurnal.unesa.ac.id/bank/jurnal/8._artikel_dwi_desi_dan_nursalim.pdf (Diakses pada Tanggal 26 Mei 2013)
- Rosnelly dan Wardoyo.2011. *Penerapan Fuzzy Multi Criteria Decision Making (Fmcdm) Untuk Diagnosis Penyakit Tropis*. UPN "Veteran" Yogyakarta
<http://repository.upnyk.ac.id/642/1/D-4.pdf>
(Diakses tanggal 26 Mei 2013)
- Sabyla, Fitrya Nadya. 2012.*Perancangan Dan Implementasi Search Engine Menggunakan Teknologi Semantic Web Pada Aplikasi Digital Library*:Diglib:Uin Malang
- Saabas, ando. 2007. *Sphider is a popular open-source web spider and search engine*.<http://www.sphider.eu>. (Diakses pada tanggal 26 September 2013)
- Sartin. 2008. *Pemilihan Supplier Bahan Baku Dengan Menggunakan Metodemulti Criteria Decision Making (Mcdm) With Promethee Dan Goal Programming Diperusahaan Azam Jaya Sidoarjo*. FTI-UPNV Jatim
- Syihab, M.Qurays, *Tafsir Al-misbah Pesan, Kesan & Keserasian Al-Quran*),

- Sulhan. 2010. *Penerapan Semantic Searching Berbasis Ontologi pada Perpustakaan Digital*: Universitas Sumatera Utara
- Sutjiati, Rosemarie.2010. *Studi akuntabilitas Pada perguruan tinggi swasta*: Universitas Kristen Maranatha Bandung
<http://repository.maranatha.edu/620/1/Studi%20Akuntabilitas.pdf> (Diakses pada tanggal 16 april 2013)
- Suteja, Bernard Renaldy, Ahmad Ashari. 2008. *Ontology e-Learning Content berbasis Web Semantic*: Universitas Gadjah Mada.
- Wang,W dan Fenton, N. 2006. *Risk and Confidence analysis for Fuzzy Multicriteria Decision Making*. Risk Analysis and Decision Research Group, Department Computer Science, Queen Mary University of London, Inggris.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.7927&rep=rep1&type=pdf> (Diakses tanggal 26 Mei 2013)
- Xu,ling dan yang, bo-yang. 2011. *Introduction to Multi-Criteria Decision Making and the Evidential Reasoning Approach*. Working Paper No. 0106 Manchester School of Management University of Manchester Institute of Science and Technology (Diakses tanggal 27 Mei 2013)
- Wellem, Theophilus.2009. *Semantic websebagai Solusi masalah dalam e-tourism di indonesia*: Yogyakarta
- Yoyon Bahtiar Irrianto.2011.*Kebijakan Pembaruan Pendidikan, Konsep, Teori dan Model*, Jakarta: PT Rajagrafindo Persada
- <http://etutorials.org/> *Advertise on eTutorials.org* (Diakses pada tanggal 18 Oktober 2013)
- <http://www.w3schools.com/rdf/default.asp>. *Web Services Tutorial* (Diakses pada tanggal 16 April 2013)