

**EKSTRAKSI DATA PADA WEBSITE *DIGITAL LIBRARY*
MENGUNAKAN ALGORITMA *DEPTH-FIRST*
SEARCH SEBAGAI METODE PENELUSURAN
GUNA PEMBUATAN *KNOWLEDGE BASE***

SKRIPSI

MUSYFIQ LUTHFI
NIM. 09650183



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK
IBRAHIM
MALANG
2015**

**EKSTRAKSI DATA PADA WEBSITE DIGITAL LIBRARY
MENGUNAKAN ALGORITMA DEPTH-FIRST
SEARCH SEBAGAI METODE PENELUSURAN
GUNA PEMBUATAN KNOWLEDGE BASE**

SKRIPSI

Diajukan Kepada :

Fakultas Sains dan Teknologi

Universitas Islam Negeri

Maulana Malik Ibrahim Malang

Untuk Memenuhi Salah Satu Persyaratan Dalam

Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :

MUSYFIQ LUTHFI

NIM : 09650183



JURUSAN TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM

MALANG

2015

**EKSTRAKSI DATA PADA WEBSITE DIGITAL LIBRARY
MENGUNAKAN ALGORITMA DEPTH-FIRST SEARCH
SEBAGAI METODE PENELUSURAN GUNA
PEMBUATAN KNOWLEDGE BASE**

SKRIPSI

Oleh:

MUSYFIQ LUTHEI
NIM. 09650183

Telah Disetujui oleh:

Dosen Pembimbing I,

Dosen Pembimbing II,

A'la Syauqi, M.Kom
NIP. 19771201 200801 1 007

Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

Mengetahui:
Ketua Jurusan Teknik Informatika,

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

**EKSTRAKSI DATA PADA WEBSITE DIGITAL LIBRARY
MENGUNAKAN ALGORITMA DEPTH-FIRST SEARCH
SEBAGAI METODE PENELUSURAN GUNA
PEMBUATAN KNOWLEDGE BASE**

SKRIPSI

MUSYFIQ LUTHFI
NIM. 09650183

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
pada Tanggal 30 Juni 2015

Susunan Dewan Penguji

Tanda Tangan

- | | | |
|--------------------|--|--------------------------|
| 1. Penguji Utama : | <u>Dr. Cahyo Crysdian</u>
NIP. 19740424 200901 1 008 | () |
| 2. Ketua : | <u>Dr. M. Amin Hariyadi, M.T</u>
NIP. 19670118 200501 1 001 | () |
| 3. Sekretaris : | <u>A'la Svauqi, M.Kom</u>
NIP. 19771201 200801 1 007 | () |
| 4. Anggota : | <u>Dr. Muhammad Faisal, M.T</u>
NIP. 19740510 200501 1 007 | () |

Disahkan Oleh:
Ketua Jurusan Teknik Informatika,

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah kepada Allah swt atas setiap nikmat dan anugerah-Nya sholawat atas Nabi besar kita Muhammad saw, ya Allah, semoga setiap langkah selalu Engkau ridhoi dengan segala rahmat-mu

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

Bapak H. Luthfi Fathorrahman dan Ibu Hj. Khozaimah, orang tua terhebat yang selalu menyayangi dan mengasihiku dalam setiap langkah hidupku.

Adikku satu-satunya Zulvaniyah

Saudaraku tercinta M.Ubaidillah S.E, Ach.Wafidi A.md, Khotib S.Fil, Roykafi S.Ag, Fahrur Razi S.Hum, Rifqiyatul Mawaddah S.Hum terima kasih atas motivasinya

Sahabat MANISKA paling mbois Fendi Pradana, S.S, Unsul Abrar, S.E, M.M, Luthfatul Qibtiyah S.S, MPd.i terimakasih atas segala-nya.

Teman-teman Jurusan Teknik Informatika 2009

Sahabat seperjuangan organisasi daerah Forum Komunikasi Mahasiswa Sumenep dan para aktivis pejuang dari Himpunan Mahasiswa Jurusan TI, Dewan Eksekutif Mahasiswa Fakultas Sains dan Teknologi, Dewan Eksekutif Mahasiswa Universitas.

Tak lupa juga sahabat pejuang-pejuang ideologi PMII Komisariat Sunan Ampel Malang, dan PMII Rayon Pencerahan Galileo Alfian S.Si, Erwanto S.Si Driana Rieke S.Si, Aunul Mubarak S.Kom, Fathurrahman S. Si, Zami Nur Yuana S. Kom, dan semua yang tak bisa saya sebutkan seluruhnya, berkat kalian aku mendapatkan pengalaman dan pembelajaran yang luar biasa.

Kepada setiap orang yang telah membantu dan telah menjadi motivator, yang selalu mendoakan dan memberi semangat.

Terima Kasih

Atas segalanya, Semoga Allah swt selalu melindungi dan menyayangi semuanya.....

Malang 07 Juli 2015

MOTTO

“ Now Or Never “

*“ Mungkin memang tak sempurna
Tapi tak ada yang lebih bernilai dari
kesalahan dibanding sembunyi dibalik
kesangsian “*

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Musyfiq Luthfi
NIM : 09650183
Fakultas : Sains dan Teknologi
Jurusan : Teknik Informatika

menyatakan bahwa “**Skripsi**” yang saya buat untuk memenuhi persyaratan kelulusan pada Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, dengan judul:

**EKSTRAKSI DATA PADA WEBSITE DIGITAL LIBRARY
MENGUNAKAN ALGORITMA DEPTH-FIRST SEARCH SEBAGAI
METODE PENELUSURAN GUNA PEMBUATAN KNOWLEDGE BASE**

adalah hasil karya saya sendiri, bukan “**duplikasi**” dari karya orang lain.

Selanjutnya apabila di kemudian hari ada “**klaim**” dari pihak lain, bukan menjadi tanggung jawab Dosen Pembimbing dan atau pihak Fakultas Sains dan Teknologi, tetapi menjadi tanggung jawab saya sendiri.

Demikian surat pernyataan ini saya buat dengan sebenarnya dan tanpa paksaan dari siapapun.

Malang, 30 Juni 2015

Hormat saya,

MUSYFIQ LUTHFI

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Puji syukur kami haturkan kehadirat Allah SWT, yang telah memberikan rahmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul **“Ekstraksi Data Pada Website Digital Library Menggunakan Algoritma Depth-First Search Sebagai Metode Penelusuran Guna Pembuatan Knowledge Base”**. Sholawat serta salam semoga tercurahkan kepada junjungan umat Islam, Nabi Agung Muhammad SAW, dengan jiwa sucinya yang penuh pengorbanan dan keikhlasan telah membimbing dan menuntun umatnya dari jalan yang penuh kegelapan menuju jalan yang penuh dengan cahaya ilmu yang diridhoi oleh Allah SWT.

Atas terselesainya penelitian ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. H. Mudjia Rahardjo, M.Si selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Ibu Dr. drh. Hj. Bayyinatul Muchtaromah, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Dr. Cahyo Crysdiyan, MCS selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak A'la Syauqi, M.Kom selaku Dosen Pembimbing yang dengan sabar telah memberikan bimbingan, arahan, serta dukungan kepada penulis selama penyusunan skripsi ini.
5. Dr. Suhartono, M.Kom selaku Dosen Wali yang telah mengarahkan, mendampingi dan memperlancar proses perkuliahan dari semester awal hingga semester akhir.
6. Seluruh Dosen Universitas Islam Negeri Maulana Malik Ibrahim Malang khususya Dosen Jurusan Teknik Informatika, PKPBA dan PKPBI yang

telah memberikan ilmu pengetahuan kepada penulis selama dibangku kuliah.

7. Kedua orang tua dan adik penulis yang selalu memberikan yang terbaik untuk penulis dan segenap keluarga besar K. Su Sahi dan KH Masyuri
8. Teman-teman Jurusan Teknik Informatika yang telah membantu penulis dalam penelitian ini.
9. Temen-teman PKPBA Kelas C6
10. Sahabat-sahabat PMII Komisariat Sunan Ampel Malang, PMII Rayon Pencerahan Galileo, Himpunan Mahasiswa Jurusan Teknik Informatika, Dewan Eksekutif Mahasiswa Fakultas Sains dan Teknologi, Dewan Eksekutif Mahasiswa Universitas, Forum Komunikasi Mahasiswa Sumenep.
11. Dan kepada seluruh pihak yang mendukung penulisan skripsi ini, yang tidak dapat disebutkan satu persatu, penulis ucapkan terima kasih yang sebesar-besarnya.

Penulis sadar bahwa kesempurnaan hanya milik Allah SWT, semoga apa yang menjadi kekurangan dapat diperbaiki oleh peneliti selanjutnya dan semoga skripsi ini bermanfaat bagi penulis khususnya dan dapat menambah pengetahuan bagi kita semua. *Amin Ya Rabbal 'Alamin...*

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 01 Juli 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	
HALAMAN PENGANTAR	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
ABSTRAK	xiii
 BAB I PENDAHULUAN	 1
Latar Belakang	1
Rumusan Masalah	5
Batasan Masalah.....	6
Tujuan Penelitian	6
Manfaat Penelitian	6
Metode Penelitian.....	6
Sistematika Penulisan	8
 BAB II KAJIAN PUSTAKA	 9
Penelitian Terkait	9
Website.....	13
Crawler.....	15
Arsitektur WebCrawler	17
Pemograman Java.....	18
MySql.....	20
Keunggulan MySQL	21
Sistem Server Database MySQL.....	21
Fungsi-fungsi MySQL	22
XML dan XML Schema.....	23
RDF dan RDF Schema.....	24
SPARQL	25
Ontology.....	26
Algoritma Depth First Search	27
 BAB III ANALISIS DAN PERANCANGAN SISTEM	 36
Obyek Penelitian	36
Alat Penelitian	36
Hardware yang Digunakan.....	36
Software yang Digunakan	36
Deskripsi Sistem.....	37
Pemodelan Data <i>Digital Library</i> Menggunakan <i>Ontology</i>	39

Membangun Aplikasi <i>Digilib Extractor</i>	41
Perancangan <i>Database</i>	45
Perancangan <i>Resource Description Framework</i> (RDF).....	46
Perancangan Aplikasi	51
Halaman Utama Aplikasi	51
Konfigurasi Database	52
Perancangan Uji Coba.....	53
 BAB IV HASIL DAN PEMBAHASAN	 54
Implementasi <i>Interface</i>	54
Tampilan Halaman Utama	54
Halaman Pengaturan Koneksi Database	55
Fitur Penentuan Jurusan	56
Fitur Penentuan Jumlah Pengambilan Data	57
Fitur Pengambilan Data (Get Data).....	58
Fitur Export ke RDF (<i>Resource Description Framework</i>).....	60
Implementasi RDF (<i>Resource Description Framework</i>)	61
Perancangan <i>Ontology</i>	62
Ekstraksi <i>Metadata Digital Library</i>	63
Rancangan <i>Resource Description Framework</i>	67
Uji Coba	70
Integrasi Pencarian dalam Al-Qur'an dan Hadits	76
 BAB V PENUTUP.....	 83
Kesimpulan.....	84
Saran.....	84
 DAFTAR PUSTAKA	 85

DAFTAR GAMBAR

Gambar	Halaman
2.1 Konsep Penelitian Sebelumnya	12
2.2 Konsep Penelitian Saat Ini	12
2.3 Arsitektur Web Crawler	19
2.4 Contoh RDF	25
2.5 Contoh SPARQL Query.....	26
2.6 Contoh Ontology	27
2.7 Diagram Pohon dari DFS	28
2.8 Pohon Pencarian (search tree).....	30
2.9 Langkah Awal	31
2.10 Langkah 1.....	32
2.11 Langkah 2.....	32
2.12 Langkah 4.....	34
2.13 Langkah 5.....	34
3.1 Blok Diagram.....	38
3.2 Activity Diagram Aplikasi	38
3.3 Hirarki is-a	40
3.4 Desain Flowchart Aplikasi.....	42
3.5 Desain Tabel <i>Digital Library</i>	46
3.6 Contoh Implementasi RDF	48
3.7 Rancangan RDF Digital Library	50
3.8 Skema <i>Graph</i> dari RDF <i>Digital Library</i>	51
3.9 Desain Halaman Awal aplikasi	52
3.10 Desain Interface Konfigurasi Database.....	52
4.1 Halaman Utama.....	55
4.2 Pengaturan Koneksi Database.....	55
4.3 Source Code Pengaturan Koneksi Database	56
4.4 Halaman Skripsi Perfakultas	56
4.5 Source code Halaman Skripsi Perfakultas.	57
4.6 Fitur Penentuan Jumlah Pengambilan Data	57
4.7 Source code Fitur Penentuan Jumlah Pengambilan Data.....	59
4.8 Fitur Get Data.....	59
4.9 Hasil Fitur Get Data	59
4.10 Source code Get Data.....	59
4.11 Fitur Export Ke RDF.....	60
4.12 Source code Fitur Export Ke RDF	61
4.13 Rancangan <i>Ontology digital library</i>	62
4.14 Teknik Penelusuran <i>Algortima Dept First Search</i>	63
4.15 Proses Ekstraksi Dokumen <i>Metadata Digital Library</i>	65
4.15 Source code Proses Ekstraksi Dokumen <i>Metadata Digital Library</i>	65
4.17 Source code penyimpanan ke database	66
4.18 Sintak RDF <i>Digial Library</i>	68
4.19 SPO dari <i>Digial Library</i>	69
4.20 Skema <i>graph</i> dari <i>Digial Library</i>	70
4.21 Proses upload RDF ke fuseki server	71
4.22 Query untuk menampilkan jurusan Teknik Informatika.....	72

4.23 Hasil Query untuk menampilkan semua skripsi jurusan Teknik Informatika.....	72
4.24 Query untuk menampilkan skripsi dari Fakultas Psikologi	73
4.25 Query untuk menampilkan semua skripsi dari Fakultas Psikologi	73



DAFTAR TABEL

Tabel	Halaman
3.1 Tabel Hasil Rancangan RDF	54
4.1 Tabel Database MySql	67
4.2 Tabel Pengujian	50



ملخص

مشفق لطفي. 2015. استخراج البيانات في المكتبة الرقمية الويسيت باستخدام خوارزمية

Depth-First Search كطريقة البحث لجعل قاعدة المعرفة (Knowledge Base).

المشرف : (1) أعلى شوقي الماجستير، (2) الدكتور مُجَّد فيصل الماجستير

الكلمات الأساسية : استخراج، Resource Description Framework، قاعدة المعرفة

المكتبة الرقمية هي مفهوم استخدام الإنترنت وتكنولوجيا المعلومات في إدارة المكتبة. كثيرة من المكتبات في إندونيسيا التي نفذت المكتبة الرقمية لدعم عمليات المكتبة اليومية. أما Resource Description Framework إعطاء معايير الفوقية التي تمكن من عملية تبادل البيانات مرنة من اتحاد شبكة ويب العالمية (World Wide Web Consortium) كوسيلة لوصف الموارد الموجودة على شبكة الإنترنت.

Resource Description Framework هو الفوقية التي تستخدم لوصف عنوان الموارد على شبكة الإنترنت. هذه الفوقية تشمل العنوان، والمؤلف، وحقوق الطبع، والنشر، والترخيص في وثيقة الويب. عنصر البيان في Resource Description Framework تتكون على الفاعل، والمسند، والموضوع. الموضوع هو الشيء الموضح ويحتوي على عنوان URI. في هذا الحال عنوان URI يمثل الموارد. المسند هو عقارات من الموارد التي توضح العلاقة بين الفاعل والموضوع. وأيضا الموضوع هو قيمة من المسند. الموضوع يحتوي على نوعين من البيانات التي تحتوي على نوع من URI.

دلت النتائج بأن تصميم Resource Description Framework يستطيع أن يستخدم كجعل knowledge base digital library جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. نتيجة query sparql القائمة قادرة على تمثيل الفاعل، والفعل، والمفعول من نتيجة Resource Description Framework ولكن نتيجة query sparql knowledge base لا تحصل على النتيجة الأقصى. هذا الحال يسبب بتشابه بيانات في قسم التربية الإسلامية للدراسات العليا والتربية الإسلامية بحيث كانت دقة النتائج المحسولة 97%.

ABSTRACT

Musyfiq Lutfi. 2015. Data Extraction on *Website Digital Library Using Depth-First Search Algorithm as Search Methods in making Knowledge Base*.

Advisors: (I) A'la Syauqi, M.Kom (II) Dr. Muhammad Faisal, M.T

Keywords: Extraction, Resource Description Framework, Knowledge Base

Digital library is the concept of the use of the Internet and information technology in library management. Many libraries in Indonesia that have implemented digital library support operations of daily library. While the *resource description framework* (RDF) provides a metadata standard that enables the flexible data exchange process by the *World Wide Web Consortium* as a way to describe the resource on the web.

RDF (*Resource Description Framework*) is metadata that uses to describe the address of the resource on the web. Metadata may include title, author, copyright, and licensing in a web document. RDF statements element are consists of a subject, predicate and object. The subject is something that describes and usually contains of the URL address. In this case, the URI address represents the resource. The predicate is the property of the resource that explains the relationship between the subjects with the object. Besides the object is the value of a predicate. The object has two types of data are object that has the type of URI.

Based on research that has been done can be concluded that the results of designing of RDF (*Resource Description Framework*) can be used as making a knowledge base digital library of UIN Maliki. *Sparql Query* represented a subject-predicate-object of the results design of RDF (*Resource Description Framework*) but the results of *Sparql Query Knowledge base* did not entirely get the maximum results. This was due to the similarity of data RDF S2 of Islamic education with Islamic education so the results accuracy obtained 97%.

ABSTRAK

Musyfiq Luthfi. 2015. Ekstraksi Data pada Website Digital Library Menggunakan Algoritma Depth-First Search Sebagai Metode Penelusuran Guna Pembuatan Knowledge Base.

Pembimbing : (I) A'la Syauqi, M.Kom (II) Dr. Muhammad Faisal, M.T

Kata Kunci : Ekstraksi, Resource Description Framework, Knowledge Base

Perpustakaan digital (*digital library*) merupakan konsep penggunaa internet dan teknologi informasi dalam manajemen perpustakaan. Banyak perpustakaan di indonesia yang sudah menerapkan perpustakaan digital untuk mendukung operasional perpustakaan sehari-hari. Sedangkan *resource description framework* (RDF) memberikan suatu standar metadata yang memungkinkan terjadinya proses pertukaran data secara fleksibel oleh *World Wide Web Consortium* sebagai cara untuk mendeskripsikan *resource* yang ada di web.

RDF (*Resource Description Framework*) merupakan metadata yang digunakan untuk medeskripsikan alamat sumber daya pada web. Metadata ini dapat berupa judul, pengarang, hak cipta, dan lisensi dalam dokumen web. Elemen pernyataan dalam RDF terdiri dari subjek, predikat dan objek. Subjek adalah sesuatu yang dideskripsikan dan biasanya berisikan alamat URI. Dalam hal ini alamat URI merepresentasikan sumber daya. Predikat merupakan properti dari sumber daya yang menjelaskan hubungan antara subjek dengan objek. Selain itu objek merupakan nilai dari sebuah predikat. Objek mempunyai dua tipe data yaitu objek yang mempunyai tipe URI.

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa hasil perancangan RDF (*Resource Description Framework*) dapat digunakan sebagai pembuatan *knowledge base digital library* UIN Maliki Malang. Hasil *query sparql* yang dilakukan telah dapat merepresentasikan subjek-predikat-objek dari hasil perancangan RDF (*Resource Description Framework*) akan teteapi hasil *query sparql knowledge base* belum seluruhnya mendapat hasil yang maksimal. Hal ini dikarenakan adanya kesamaan data RDF pada jurusan S2 Pendidikan Agama Islam dengan Pendidikan Agama Islam sehingga hasil akurasi yang diperoleh adalah 97 %.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi Informasi adalah suatu teknologi yang digunakan untuk mengolah data, Termasuk memproses, mendapatkan, menyusun, menyimpan, memanipulasi data dalam berbagai cara untuk menghasilkan informasi yang berkualitas, yaitu informasi yang relevan, akurat dan tepat waktu, yang digunakan untuk keperluan pribadi, bisnis, dan pemerintahan dan merupakan informasi yang strategis untuk pengambilan keputusan. Teknologi ini menggunakan seperangkat komputer untuk mengolah data, sistem jaringan untuk menghubungkan satu komputer dengan komputer yang lainnya sesuai dengan kebutuhan, dan teknologi telekomunikasi digunakan agar data dapat disebar dan diakses secara global.

Pertumbuhan jumlah situs web (*website*) di Internet berdasarkan hasil *survey* dari Netcraft (2013) menunjukkan peningkatan pesat dari 18 juta *website* pada tahun 2000 menjadi 716 juta *website* pada tahun 2013. Berbagai jenis informasi yang ada di Internet ditransmisikan secara tidak terstruktur melalui *website*. Informasi yang ada pada *website* ditampilkan dalam bentuk halaman-halaman web (*web pages*), secara umum terdiri dari beberapa bagian, antara lain *content* utama, iklan, navigasi, dan informasi tambahan lainnya.

Seiring dengan meningkatnya pertumbuhan jumlah *website* tersebut, tentunya semakin berkembang pula kemampuan kita dalam mengumpulkan dan mengolah data. Penggunaan sistem komputerisasi dalam berbagai bidang baik itu

dalam bidang pendidikan, transaksi-transaksi bisnis, maupun untuk kalangan pemerintah dan sosial, telah menghasilkan data yang berukuran sangat besar.

Dengan diperolehnya informasi-informasi yang berguna dari data-data yang ada, hubungan antara item dalam transaksi, maupun informasi-informasi yang potensial, selanjutnya dapat diekstrak dan dianalisa dan diteliti lebih lanjut dari berbagai sudut pandang. Informasi yang ditemukan ini selanjutnya dapat diaplikasikan untuk manajemen, melakukan query processing, pengambilan keputusan dan lain sebagainya. Dengan semakin berkembangnya kebutuhan akan informasi-informasi, semakin banyak pula bidang-bidang yang menerapkannya. Dengan banyaknya publikasi dokumen-dokumen tersebut di internet, Sehingga dengan kemudahan mengakses dokumen dan mendapatkannya, banyak dokumen-dokumen yang beredar di internet tersebut kemudian dijadikan bahan acuan atau referensi oleh pihak-pihak yang berkaitan di dunianya.

Untuk mengetahui keterkaitan suatu dokumen yang dijadikan referensi oleh dokumen lainnya, maka diperlukan suatu cara untuk melihat referensi-referensi yang digunakan oleh suatu dokumen tersebut.

Dengan berdasarkan dokumen yang tersebar di internet, maka untuk dapat mengetahui isi dari dokumen tersebut, khususnya bagian referensi dan mencari keterkaitannya, maka dibutuhkan suatu cara untuk dapat membaca isi dari dokumen yang berada pada halaman suatu web. Cara tersebut umumnya disebut Web crawler.

Web crawler merupakan program yang mengumpulkan informasi yang akan ditempatkan pada basis data. Sebuah web crawler bertugas menelusuri web dan

mengumpulkan dokumen-dokumen di dalamnya. Selanjutnya web crawler akan mengurutkan dokumen-dokumen tersebut dan membangun sebuah daftar indeks. Tujuan pengurutan tersebut adalah untuk menentukan seberapa penting suatu dokumen dan untuk menduga URL-URL yang mungkin penting sehingga URL tersebut perlu ditelusuri terlebih dahulu (Junghoo Cho dkk, 1998).

Sejumlah metode penelusuran berkaitan dengan pengurutan tersebut salah satunya adalah *algoritma depth first search* (DFS). Pada metode penelusuran *depth first search*, web crawler akan menelusuri dokumen-dokumen metadata yang bersifat lokal terlebih dahulu. Kemudian web crawler akan menelusuri dokumen-dokumen pada situs lain. Salah satu contoh metode penelusuran seperti ini adalah penelusuran berdasarkan banyaknya jumlah backlink.

Universitas Islam Negeri Maulana Malik Ibrahim Malang, sebagai salah satu langkah untuk menuju *World Class University*, tentunya harus dilakukan pengembangan infrastruktur, meliputi gedung-gedung perkuliahan, masjid, mahad, perpustakaan, laboratorium, pusat studi, *students centre*, dan berbagai gedung untuk fasilitas penunjang lainnya. Perpustakaan merupakan salah satu bangunan yang perlu dikembangkan. Selama ini perpustakaan yang ada di kampus UIN Malang belum cukup untuk memenuhi kebutuhan belajar guna menghasilkan lulusan yang mampu berperan dalam keilmuan internasional (Sugeng, 2014).

Perpustakaan merupakan tempat, gedung atau ruang yang disediakan untuk menyimpan buku-buku bacaan, artikel, jurnal maupun skripsi dari mahasiswa, sehingga mahasiswa dapat menggunakannya untuk mendapatkan literatur yang diperlukan sebagai penunjang kegiatan perkuliahan di kampus ini. Perpustakaan

memiliki banyak fungsi terutama dalam proses belajar mengajar, salah satunya adalah membaca, karena membaca merupakan salah satu cara untuk memperoleh pengetahuan, membaca dapat menambah wawasan dan memperkaya diri dengan ilmu. Terlebih ketika hendak menulis skripsi, tentunya perlu banyak membaca skripsi terdahulu sebagai referensi.

Selain pembangunan secara fisik, pengembangan teknologi informasi tentunya mempunyai peranan penting. Pengembangan sistem informasi diharapkan dapat memperluas dan mempercepat akses informasi seluruh warga kampus. Sistem informasi di perpustakaan perlu dikembangkan agar diperoleh akses yang luas, mudah, dan cepat untuk mencari berbagai literatur dan sumber belajar seluruh civitas akademika UIN Malang. Untuk itu, dibuatlah sebuah perpustakaan digital atau *digital library* sehingga memudahkan untuk mendapatkan informasi mengenai buku ataupun skripsi yang tersimpan di perpustakaan pusat UIN Malang dengan cepat.

Perpustakaan digital (*digital library*) merupakan konsep penggunaa internet dan teknologi informasi dalam manajemen perpustakaan. Banyak perpustakaan di indonesia yang sudah menerapkan perpustakaan digital untuk mendukung operasional perpustakaan sehari-hari.

Sedangkan *resource description framework* (RDF) memberikan suatu standar metadata yang memungkinkan terjadinya proses pertukaran data secara fleksibel oleh World Wide Web Consortium sebagai cara untuk mendeskripsikan *resource* yang ada di web.

Meningkatnya kebutuhan untuk saling berbagi informasi dan keterbatasan tempat penyimpanan menjadikan *resource sharing* sebagai hal yang cukup krusial bagi perpustakaan digital. Namun, *resource sharing* tidaklah mudah untuk diimplementasikan. Ini disebabkan perbedaan standar yang digunakan oleh masing-masing perpustakaan. Selanjutnya, dalam penelitian ini akan dijabarkan bagaimana menggunakan RDF sebagai jembatan untuk tukar menukar data antar metadata yang berbeda.

Dengan memanfaatkan hasil perancangan *resource description framework* dengan menggunakan proses ekstraksi halaman skripsi serta *Dept First Search* sebagai metode penelusuran skripsi *digital library* diharapkan dapat menjadi salah satu solusi untuk meningkatkan efektifitas sistem dan mengatasi dari permasalahan yang ada dikemudian hari. Sehingga hasil proses yang dihasilkan dapat membantu universitas dalam penyempurnaan sistem selanjutnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana membangun sebuah aplikasi ekstraksi pada perancangan *resource description framework* untuk pembuatan *knowledge base digital library* UIN Maulana Malik Ibrahim Malang?
- b. Seberapa baik perancangan *resource description framework* dalam pembuatan *knowledge base digital library* UIN Maulana Malik Ibrahim Malang?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut:

- a. Sumber data berasal dari *digital library* Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- b. Penelitian ini difokuskan pada pembuatan *resource description framework digital library* UIN Maulana Malik Ibrahim Malang.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah

- a. Membangun aplikasi ekstraksi data *digital library* untuk perancangan *resource description framework* guna pembuatan *knowledge base digital library* UIN Maulana Malik Ibrahim Malang.
- b. Mengukur nilai akurasi hasil pembuatan *knowledge base digital library* dengan menggunakan dari hasil perancangan *resource description framework*.

1.5 Manfaat Penelitian

Manfaat dari pembuatan aplikasi ini adalah diharapkan agar hasil dari perancangan *resource description framework* ini dapat mempermudah universitas dalam pembuatan *knowledge base digital library*.

1.6 Metode Penelitian

Penelitian ini akan dilakukan dalam beberapa tahap, berikut ini penjelasan dari tahap penelitian yang akan dilakukan:

a. Studi Literatur

Pada tahap ini dilakukan pencarian sekaligus pemahaman terkait teori ekstraksi dan *resource description framework*. Literatur yang digunakan berupa buku, paper, jurnal, dan tugas akhir mahasiswa jurusan Teknik Informatika.

b. Perumusan Masalah dan Penyelesaiannya

Pada tahap ini meliputi perumusan masalah, batasan masalah, dan penyelesaian masalah tersebut.

c. Perancangan dan Desain Sistem

Pada tahap ini dilakukan perancangan dan desain sistem untuk menerapkan penyelesaian masalah.

d. Pembuatan Sistem

Pada tahap ini dilakukan pembuatan sistem sesuai dengan perancangan dan desain sistem yang telah ada.

e. Uji Coba dan Evaluasi Hasil

Tahap ini merupakan uji coba perancangan *resource description framework* pada sistem yang telah dibuat.

f. Penyusunan Tugas Akhir

Pada tahap ini dilakukan penulisan tugas akhir yang merupakan dokumentasi dari konsep atau teori penunjang, perancangan dan desain sistem, pembuatan sistem, dokumentasi dari uji coba dan analisis, serta kesimpulan dan saran.

1.7 Sistematika Penulisan

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut:

BAB I Pendahuluan

Pendahuluan berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

BAB II Kajian Pustaka

Kajian pustaka berisi beberapa teori yang mendasari dalam penyusunan tugas akhir ini, serta beberapa penelitian terdahulu yang berkaitan dengan penelitian ini. Adapun yang dibahas dalam bab ini adalah dasar teori yang berkaitan dengan *crawler* dan *resource description framework*.

BAB III Metode Penelitian

Pada bab ini dijelaskan metode yang digunakan dan tahapan-tahapan pembuatan sistem untuk menyelesaikan masalah.

BAB IV Hasil dan Pembahasan

Menjelaskan tentang implementasi dari perancangan *resource description framework* dalam pembuatan *knowledge base digital library*.

BAB V Penutup

Berisi kesimpulan dari penelitian ini dan saran untuk penelitian selanjutnya.

BAB II

KAJIAN PUSTAKA

2.1 Penelitian Terkait

Pada tahun 2012 terdapat penelitian dilakukan oleh Himmatul Azizah-Sarwosri, S.Kom., M.T., Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng tentang Ekstraksi Metadata Dokumen Software Requirement Specification (SRS). Dokumen Software Requirements Specification (SRS) merupakan sebuah deskripsi lengkap dari behavior sebuah sistem yang akan dikembangkan. Dokumen ini berisi rincian kebutuhan fungsionalitas dan nonfungsionalitas. Dokumen ini dianalisa oleh pengembang perangkat lunak yang kemudian digunakan oleh para pengembang perangkat lunak untuk dapat menciptakan perangkat lunak yang sesuai dengan kebutuhan pengguna. Pada kenyataannya, pengembang perangkat lunak terkadang sulit untuk menganalisa sebuah dokumen SRS yang digunakan sebagai pedoman saat pembuatan perangkat lunak. Oleh karena itu, diperlukan adanya perangkat lunak yang dapat membantu developer menganalisa dokumen SRS. Dalam penelitian ini hasil yang telah dicapai berupa Membuat sebuah kakas bantu untuk mengekstraksi sebuah dokumen SRS, Hasil fitur yang dapat diekstraksi adalah kebutuhan fungsional, kebutuhan non fungsional, use case, dan aktor. Keakurasian sistem dalam mengekstraksi kebutuhan fungsional adalah 100%, untuk kebutuhan non fungsional 0%, untuk use case 100%, dan untuk aktor 4,31%. Waktu yang dibutuhkan untuk mengekstraksi sebuah dokumen SRS ditentukan dari besarnya ukuran file dokumen yang ingin diproses. Use

case dan aktor dari sebuah perangkat lunak yang akan dibangun dapat ditemukan pada bagian kebutuhan fungsional pada dokumen SRS.

Penelitian berikutnya yaitu Penelitian oleh Ferdila tentang Aplikasi Web Semantik Untuk Pencarian Materi Perkuliahan. Penelitian ini bertujuan mengelola secara semantis informasi materi perkuliahan di Gunadarma dengan menggunakan staffsite gundarma. Pada halaman staffsite dapat melakukan pencarian materi perkuliahan sesuai dengan nama dosen yang mengajar suatu mata kuliah. sehingga pengguna dapat melakukan pencarian berdasarkan persepsi dan tingkat pengetahuan yang dimilikinya untuk mencari mata kuliah.

Dalam penelitian ini, pencarian materi perkuliahan dapat dilakukan 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. Penggunaan teknologi semantik web pada aplikasi ini bertujuan agar representasi data yang tersedia dapat diubah kedalam format dapat dimengerti oleh mesin. Sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan informasi dengan lebih baik. Pemilihan RAP sebagai penghubung antara ontologi RDF yang ada dengan tampilan yang ada membuat manipulasi keluaran untuk user lebih mudah dilakukan sehingga dapat menghasilkan keluaran yang mudah dimengerti oleh pengguna.

Pengembangan *semantic search* pada *history web browser*. Sehingga menghasilkan halaman web yang relevan pada *query* yang dicari. Browser yang digunakan adalah *Mozilla Firefox*. *History* dari user disimpan dalam *database*

yang diakses menggunakan *PHP*. *Ontology* dibuat dari *browsing history*, yang kemudian diuraikan untuk memproses serach query yang dimasukkan secara *semantic* sehingga menghasilkan *output* yang relevan. Penelitian ini menggunakan 3 modul yaitu: *Web Content Extraction*, *Semantic Knowledge Base* dan *Reasoner*. *Web Content Extraction* ini dibangun menggunakan *python*, modul ini akan mengekstrak *history* dari *web browser* dan menyimpannya ke dalam *database* tiap 15 menit, sehingga dapat diakses melalui *PHP*.

Semantic Knowledge Base dibangun menggunakan *Protege-OWL editor*. Modul ini terdiri dari dua *class* yaitu: *AppleInc* dan *AppleFruit*. Kemudian *URL* yang didapat dari modul 1 ditambahkan ke dalam *subclass* dari modul 2. *Ontology* diquery menggunakan *DL query*. *Query* ini mendapatkan informasi dari *ontolgy* yang telah dibuat pada modul sebelumnya. *Ontolgy* ini disimpan sebagai file *OWL*, dan diquery menggunakan aplikasi *PHP*. Aplikasi menampilkan hasil *query user* terhadap halaman web yang pernah dikunjungi *user*. Aplikasi ini hanya bekerja pada satu browser saja dan akan lebih efektif jika aplikasi ini berbentuk *add ons* atau *plugins*. (Monica Shekhar, dkk. 2013)

Terdapat juga pengembangan *semantic search engine* untuk informasi biologi modern dan obat tradisional Cina. *Bio TCM-SE* ini terdiri dari 3 modul, yaitu: *Ontology Modelling Module*, *Data Integration Module* dan *Semantic Search Module*. *Ontology* di sini berfungsi mendefinisikan relasi antar *subclass*. Setelah membuat *ontology*, selanjutnya membuat data integrasinya. Pertama mendownload konten dari berbagai sumber data kemudian dikonvert menjadi format *RDF* sesuai dengan skema *ontology* yang telah dibuat. Setelah itu menormalisasi *URI* dan mengubah *RDF* menjadi sebuah *graph* pengetahuan

biomedical yang besar, sehingga *user* bisa mendapatkan hasil *query* yang akurat. (Xi Chen, dkk. 2014)

Pada penelitian sebelumnya, Fitria Nadya Sabyla merancang pencarian buku pada *digital library* UIN Malang dengan memanfaatkan *RDF* sebagai tempat penyimpanan data dan *framework exhibit* untuk mengekstrak data yang tersimpan dalam *RDF*, *user* dapat menentukan jurusan apa yang akan ditampilkan dengan menggunakan *combo box* yang disediakan di bagian kanan sistem. Namun tidak jauh berbeda dengan pencarian yang ada saat ini *user* tetap harus memikirkan kata kunci yang tepat untuk mendapatkan hasil yang sesuai dan pencarian hanya bisa dilakukan untuk mencari judul buku saja.



Gambar 2.1 Konsep penelitian sebelumnya (Fitria Nadya, 2012)



Gambar 2.2 Konsep Penelitian Saat ini (Fitria Nadya, 2012)

Penelitian yang akan dilakukan yakni pencarian skripsi pada *digital library* UIN Malang dengan memanfaatkan *SPARQL* yang sebelumnya dilakukan proses pengolahan kata sehingga menghasilkan *endpoint* yang tepat serta *OWL* sebagai penyimpanan *n-triple store*. Sehingga pencarian diharapkan lebih kaya dari sistem pencarian sebelumnya. *User* tidak perlu lagi memasukkan kata kunci yang benar-benar sama untuk mendapatkan hasil pencarian yang diinginkannya.

2.2 Website

Website adalah suatu layanan untuk menghubungkan pengguna komputer dengan *internet*. *Website* menyediakan informasi bagi pengguna komputer yang terhubung ke *internet*. Pengetahuan berkembang dengan cepat melalui *website* hal ini karena manusia mampu mengakses informasi, menggabungkan, dan menciptakan inovasi baik dari sosial maupun teknologi yang dapat dimanfaatkan oleh umat manusia. Arsitektur *web* ini pertama kali dikembangkan untuk tujuan militer dalam proyek yang dinamakan DARPA yang menghubungkan komputer pertama kali. Perkembangan ini kemudian dilanjutkan oleh Tim Berners-Lee dari Massachusetts Institute of Technology (MIT) dengan memperkenalkan apa yang disebut sebagai *website*.

Standar *web* 1.0 merupakan bentuk *web* yang paling awal. Hal yang disajikan dalam *web* ini masih bersifat statis dan cenderung hanya bersifat informatif. Situs *web* statis merupakan situs *web* yang memiliki isi tidak dimaksudkan untuk diperbarui secara berkala sehingga pengaturan ataupun pemutakhiran isi *website* tersebut dilakukan secara manual. Layanan *internet* waktu itu masih berupa *website* statis yang saling dihubungkan dengan *hyperlink*. Umumnya *website* berformat brosur online, *website* yang menyampaikan informasi satu arah yang berbentuk profile, berita, toko online, email, dll. *Web* waktu itu didominasi *website* yang didesain menggunakan tabel dan *flash*.

Pada pengembangan *web* kedua (*Web* 2.0) pengunjung mulai dapat melakukan interaksi yang diatur oleh sistem yang ada. *Web* 2.0 sendiri merupakan sebuah istilah yang pertama kali dicetuskan pada tahun 2003 oleh O'Reilly Media, dan dipopulerkan pada konferensi *web* 2.0 pertama di tahun 2004. Dalam standar

web 2.0, *web* sudah merupakan media interaksi antar sesama pengguna. Bentuk yang menjadi khas pada generasi ini adalah *web* bukan hanya merupakan sumber bacaan dan mencari informasi namun juga sebagai bagian dari interaksi sosial.

Web 3.0 adalah generasi ketiga dari layanan internet berbasis *web*. Konsep *Web 3.0* pertama kali diperkenalkan pada tahun 2001, saat Tim Berners-Lee, penemu *World Wide Web*, menulis sebuah artikel ilmiah yang menggambarkan *Web 3.0* sebagai sebuah sarana bagi mesin untuk membaca halaman-halaman *Web*. Hal ini berarti bahwa mesin akan memiliki kemampuan membaca *Web* sama seperti yang manusia dapat lakukan sekarang ini. *Web 3.0* berhubungan dengan konsep *semantic web*, yang memungkinkan isi *web* dinikmati tidak hanya dalam bahasa asli pengguna, tapi juga dalam bentuk format yang bisa diakses oleh agen-agen *software*. Beberapa ahli bahkan menamai *Web 3.0* sebagai *Web Semantik* itu sendiri.

Keunikan dari *Web 3.0* adalah konsep dimana manusia dapat berkomunikasi dengan mesin pencari. Kita bisa meminta *Web* untuk mencari suatu data spesifik tanpa bersusah-susah mencari satu per satu dalam situs-situs *Web*. *Web 3.0* juga mampu menyediakan keterangan-keterangan yang relevan tentang informasi yang ingin kita cari, bahkan tanpa kita minta. *Web 3.0* menawarkan metode yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data *online*. *Web 3.0* juga memungkinkan fitur *Web* menjadi sebuah sarana penyimpanan data dengan kapasitas yang luar biasa besar. Walaupun masih belum sepenuhnya direalisasikan, *Web 3.0* telah memiliki beberapa standar operasional untuk bisa menjalankan fungsinya dalam menampung *metadata*, misalnya

Resource Description Framework (RDF) dan *Web Ontology Language (OWL)* (Berners-Lee dkk, 2001).

2.3 Crawler

Crawler adalah program otomatis khusus yang mampu mengikuti link-link yang ditemukan pada halaman-halaman web yang menuntun spider dalam menentukan web yang lain untuk dikunjungi kemudian (Feri Sulianta, 2008).

Buku lain menyebutkan Crawler merupakan program yang dimiliki oleh search engine untuk melacak dan menemukan link yang terdapat pada setiap halaman yang ditemui (Febrian, 2007).

WebCrawler merupakan program yang mengumpulkan informasi yang akan ditempatkan pada basis data. WebCrawler disebut juga web spider, web robot, atau dalam komunitas FOAF disebut web scutter adalah program atau script yang secara otomatis mencari World Wide Web dalam suatu metode atau cara yang otomatis. Proses ini disebut web crawling atau spidering. Beberapa site, khususnya search engine menggunakan spidering untuk memberikan data yang up-to-date. Web crawler umumnya digunakan untuk membuat penggandaan dari halaman yang dikunjungi dalam pemrosesan terakhir oleh search engine yang akan mengindeks halaman yang di download untuk memberikan pencarian yang lebih cepat. Crawler juga dapat digunakan secara otomatis untuk memelihara task pada website seperti pengecekan link atau validasi kode html.

Web crawler adalah layanan web yang membantu user dalam pelayarannya di web dengan otomatis tugasnya telah terhubung. Membuat pencarian indeks web dan memenuhi pencarian query dari indeks. Secara konseptual, Web crawler

adalah node dalam web graph yang memiliki hubungan ke beberapa site di internet, singkatnya penghubung antara user dan tujuannya. Simplikasi dari penggunaan web adalah penting dalam beberapa alasan.

Pertama, Web Crawler menyimpan waktu user ketika melakukan pencarian sebagai ganti dari percobaan menerka penghubung dari halaman ke halaman. Bahkan, user akan melihat hubungan yang tidak jelas antara halaman yang ditampilkan dan halaman yang dicari. Sebagai contoh, akan ditampilkan halaman dalam satu topik dan meminta halaman pada topik komplit berbeda, salah satunya tidak terhubung dari lokasi yang ada. Dalam beberapa kasus, dengan melangkah ke Web Crawler selain menggunakan alamat ini atau tombol pada browser pencarian akan lebih mudah menentukan lokasi halaman tujuan. Penghematan adalah hal yang sangat penting dengan menambahkan ukuran dan cakupan Web (Matrix Information and Directory Services Inc, 2002).

Kedua, Simplikasi Web Crawler dari penggunaan web membuat web mudah digunakan dan memiliki tool yang berguna. Navigasi web pencarian kata kunci bahkan lebih intuitif daripada mencoba menggunakan Uniform Resource Locator (URL) untuk mengidentifikasi halaman web secara langsung. Jika user memiliki pengalaman yang baik maka akan lebih menggunakan web, dan mengulangi penggunaan yang diikuti penambahan suatu medium.

Search engine seperti web crawler mempunyai peranan dalam melanjutkan penyederhanaan dan pertumbuhan web. Sehingga, web crawler menyediakan beberapa konteks untuk pencarian queri tertentu. Dengan isu well-form queri, pencarian dapat menemukan informasi yang lebih luas mengenai topik tertentu

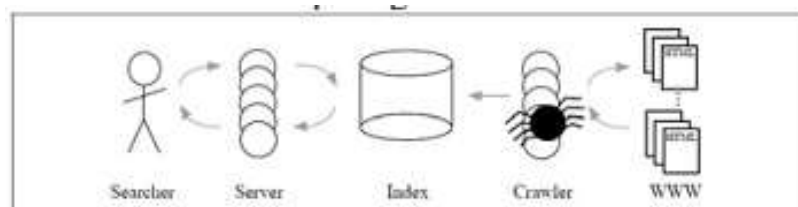
dan dapat menggunakan informasi lebih jauh dalam memperbaiki tujuan ini. Pencari sering mengisukan queri besar yang disaring seperti yang lebih banyak dipelajari tentang subjek yang diharapkan.

Sistem pencarian berbasis semantik menggunakan metadata berupa Resource Description Framework (RDF) sebagai sumber informasinya. Website Parser Template (WPT) XML berbasis open format yang menyediakan struktur HTML dari suatu halaman website. Format WPT memungkinkan web crawler untuk men-generate RDF semantic web pada suatu halaman web. Pada makalah ini dipaparkan pemanfaatan webcrawler yang disajikan berdasar website parser template untuk membangun repositori metadata pada sistem pencarian publikasi ilmiah berbasis semantik.

2.4 ARSITEKTUR WEBCRAWLER

Dalam implementasi web crawler, pada perspektif pengguna web khususnya web keseluruhan tidak terdapat elemen sisi client di luar browser yang dianggap perlu. Layanan ini terdiri atas 2 bagian penting, yaitu :

- crawling, proses menemukan dokumen dan konstruksi indeks.
- Serving, proses menerima queri dari pencari dan menggunakan indeks untuk menentukan hasil yang diinginkan.



Gambar 2.3 Arsitektur WebCrawler (Brain Pinkerton, 2000)

Walaupun implementasi Web Crawler berkembang setiap waktu, komponen dasarnya tetap sama. Hanya cakupan, detil, dan waktu relatifnya mengalami perubahan. Crawler tidak hanya memiliki strategi crawling yang baik, tetapi juga harus memiliki arsitektur optimisasi yang tinggi.

Shkopenyuk dan Suel (2002), memberi uraian bahwa : “Selama hal ini cukup mudah untuk membangun crawler lambat yang mendownload beberapa halaman per detik untuk periode waktu yang lebih singkat, membangun sistem performance tinggi dapat mendownload beribu-ribu miliar halaman selama beberapa minggu yang saat ini menjadi sejumlah tantangan dalam sistem desain, efisiensi I/O dan jaringan, ketahanan dan kemampuan mengelolanya.

Web crawler adalah bagian inti dari suatu search engine, dan detail dari algoritma dan arsitektur tetap menjadi rahasia bisnis. Ketika desain crawler di publish sering terdapat hal-hal penting karena tidak adanya detail yang mencegah hal lainnya dari reproduksi pekerjaan. Selalu timbul hal tentang “search engine spamming” yang mencegah search engine utama dari dipublikasikannya algoritma peringkatnya.

2.5 Pemrograman Java

Bahasa pemrograman adalah suatu istilah yang menjelaskan bahasa yang dapat dipahami oleh komputer. Dalam dunia perkomputeran dikenal dua jenis bahasa pemrograman, yaitu bahasa pemrograman tingkat tinggi dan bahasa pemrograman tingkat rendah.

Sebuah bahasa pemrograman atau bahasa komputer di artikan sebagai teknik komunikasi yang sudah distandarisasi untuk menyatakan instruksi-instruksi

kepada komputer. Bahasa pemrograman adalah kesatuan aturan sintaksis dan semantik yang digunakan untuk mendefinisikan sebuah program (Bado Saputro, 2012).

Sebuah bahasa pemrograman memungkinkan seorang programmer untuk memutuskan secara tepat apa yang harus dilakukan oleh komputer pada saat itu, cara mengatur alur transfer data, input-output, dan kegiatan-kegiatan lain yang harus dilakukan dalam berbagai keadaan.

Bahasa pemrograman tingkat rendah merupakan bahasa pemrograman yang sulit dimengerti oleh manusia, sedangkan bahasa pemrograman tingkat tinggi lebih mudah dipahami oleh manusia. Contoh dari bahasa pemrograman tingkat rendah adalah Assembly, dan contoh bahasa pemrograman tingkat tinggi adalah C dan Pascal.

Bahasa pemrograman java adalah teknologi yang pertama kali diperkenalkan oleh Sun Microsystems sekitar tahun 1990an. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Banyak orang menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman (Putu Yoga, 2010).

Teknologi Java memiliki tiga komponen penting yang menopang Java itu sendiri, yaitu :

1. Programming-language specification
2. Application-programming interface

3. Virtual-machine specification

Bahasa Java bisa dikategorikan sebagai bahasa pemrograman berorientasi objek, pemrograman terdistribusi dan bahasa pemrograman multithreaded . Objek Java dispesifikan dengan membentuk kelas-kelas. Untuk masing-masing kelas Java, kompiler Java mengeluarkan sebuah file keluaran arsitektur netral yang akan berjalan pada berbagai implementasi dari Java Virtual Machine (JVM). Pada awalnya Java sangat digemari oleh para programmer, karena Java mendukung untuk applets , dimana program dengan akses sumber daya terbatas dapat berjalan dalam sebuah web browser. Java juga mampu menyediakan dukungan level tingkat tinggi untuk networking dan objek terdistribusi. Java juga dianggap sebagai sebuah bahasa yang mempunyai tingkat keamanan yang tinggi. Tampilan ini pada khususnya hanya menganggap bahwa sebuah program Java bisa mengeksekusi sebuah jaringan terdistribusi. Bahasa Java pada jaman ini termasuk bahasa pemrograman yang relatif mudah untuk dipelajari karena banyaknya contoh dan konsep yang beredar luas, baik berupa buku panduan belajar java maupun yang sudah dipublikasikan di internet. Tetapi program yang dibuat dalam bahasa Java juga relatif lebih banyak membutuhkan waktu saat di eksekusi (lebih lama), hal ini dikarenakan untuk menjalankan programnya dibutuhkan JVM (perantara antar program dan sistem operasi).

2.6 MySQL

MySQL adalah sebuah perangkat lunak database (basis data) sistem terbuka yang sangat terkenal di kalangan pengembang sistem database dunia yang digunakan untuk berbagai aplikasi terutama untuk aplikasi berbasis Web. MySQL mempunyai fungsi sebagai Structured Query Language (SQL) yang dimiliki

sendiri dan telah diperluas. MySQL umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi yang dinamis dan powerfull.

2.6.1 Keunggulan MySQL

Ada beberapa keunggulan dari MySQL, diantaranya adalah :

- MySQL merupakan program yang multi-threaded, sehingga dapat dipasang pada server yang memiliki multi-CPU.
- Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL API.
- Bekerja pada berbagai platform. tersedia berbagai versi untuk berbagai sistem operasi.
- Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
- Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
- Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

2.6.2 Sistem Server Database MySQL

Sistem database MySQL memiliki sistem keamanan dengan tiga verifikasi yaitu username, password dan host. Verifikasi host memungkinkan untuk membuka keamanan di localhost, tetapi tertutup bagi host lain (bekerja di lokal komputer). Sistem keamanan ini ada di

dalam database mysql dan pada tabel user. Proteksi juga dapat dilakukan terhadap database, tabel, hingga kolom secara terpisah (Abdul Kadir, 2008).

2.6.3 Fungsi-fungsi MySQL

MySQL memiliki fungsi-fungsi standar SQL dan beberapa kemampuan tambahan. Secara lengkap dokumentasi ini terdapat pada manual MySQL. Namun demikian pada bagian ini perlu disajikan beberapa fungsi SQL yang sering digunakan dalam halaman Web.

- a. `SELECT <column,...> FROM <table_name,...> WHERE <where_definition> ORDER BY <column, ...> [ASC|DESC]`. Fungsi ini berfungsi untuk memilih atau mengambil data dari sebuah tabel dengan kolom yang telah ditentukan kemudian dipanggil kolom yang diinginkan.
- b. `INSERT INTO <table_name> VALUES(<list_of_data>)`. Fungsi ini berfungsi untuk memasukkan data kedalam sebuah tabel dengan nilai atau data yang diinginkan.
- c. `UPDATE <table_name> SET column=<expression> WHERE <where_definition>`. Fungsi ini berfungsi untuk mengganti data pada sebuah tabel dengan data yang diinginkan berdasarkan syarat yang diinginkan.
- d. `DELETE FROM <table_name> WHERE <where_definition>`. Fungsi ini berfungsi untuk menghapus data pada sebuah tabel berdasarkan syarat yang diinginkan.

2.7 XML dan XML Schema

Extensible Markup Language (XML) merupakan bahasa yang didesain untuk mempermudah pengiriman dokumen melalui *web*. Berbeda dengan *Hypertext Markup Language (HTML)*, *XML* memungkinkan penggunaanya untuk mendefinisikan *custom tag* sendiri. *XML* didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama *XML* adalah data yang jika diolah bisa memberikan informasi. *XML* menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, *XML* dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar.

Sedangkan *XML Schema* merupakan bahasa yang digunakan dalam pendefinisian sekumpulan skema atau aturan yang harus dipatuhi oleh dokumen *XML*. XSD (XML Schema Definition), sebuah rekomendasi dari World Wide Web Consortium (W3C), untuk menentukan bagaimana menjelaskan secara resmi unsur-unsur dalam dokumen Extensible Markup Language (XML). Hal ini dapat digunakan oleh programmer untuk memverifikasi setiap bagian dari konten item dalam dokumen. Seperti semua bahasa skema XML, XSD dapat digunakan untuk mengekspresikan seperangkat aturan dokumen XML yang harus sesuai agar dapat dianggap "sah" menurut skema itu. Namun, tidak seperti kebanyakan bahasa skema lainnya, XSD juga dirancang dengan tujuan agar penentuan validitas dokumen yang akan menghasilkan kumpulan informasi mengikuti tipe data tertentu.

2.8 RDF dan RDF Schema

RDF singkatan dari *Resource Description Framework*, awalnya diciptakan pada awal tahun 1999 oleh W3C sebagai standar untuk pengkodean *metadata*. Nama *Resource Description Framework*, secara resmi diperkenalkan sesuai spesifikasi dokumen W3C sebagai standar. *Resource Description Framework* (*RDF*) merupakan metode umum untuk memodelkan informasi yang dibuat oleh W3C dengan ide dasar bagaimana agar dapat membuat pernyataan mengenai sebuah *resource web* dalam bentuk ekspresi “Subjek-Predikat-Objek” (yang selanjutnya disebut dengan istilah *N-Triple*). Subjek disini mengacu pada *resource* yang ingin dideskripsikan, sedangkan predikat menggambarkan kelakuan atau karakteristik dari *resource* tersebut dan mengekspresikan hubungan antara subjek dengan objek.

Pendeskripsian *resource-resource* inilah yang dikemukakan oleh W3C sebagai komponen utama untuk perangkat lunak dalam menyimpan, menukar, dan menggunakan informasi sehingga dapat dibaca oleh mesin yang selanjutnya didistribusikan melalui web.

RDF Schema merupakan kamus data atau *vocabulary* yang digunakan untuk mendeskripsikan *properties* dan *classes* dari *resources RDF*. *RDFS* merupakan singkatan dari *RDF Schema*. Berbagai singkatan seperti *RDF (S)*, *RDF-S*, atau *RDF / S* dapat digunakan, dan ini semua mengacu pada *RDF Schema*. Sebagai standar W3C, versi awalnya diterbitkan oleh W3C pada bulan April 1998. Dengan perubahan standar *RDF*, W3C merilis final rekomendasi *RDFS* pada Februari 2004, dan hal ini termasuk dalam enam dokumen yang diterbitkan sebagai

spesifikasi *RDF* yang telah diperbarui. Contoh penggunaan *RDF* ditunjukkan pada

Gambar 2.4.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://lib.uin-malang.ac.id/ontologies/Library.owl#"
xmlns:Library="http://lib.uin-malang.ac.id/ontologies/Library.owl#"
<rdf:Description
  rdf:about="http://lib.uin-malang.ac.id/ontologies/Library.owl">
  <rdf:type resource:
    " http://lib.uin-malang.ac.id/ontologies/Library.owl#advisor10f"/>
  <Library:domain>Lecture</Library:domain>
```

Gambar 2.4 Contoh RDF (Umar Faruq, 2015)

2.9 SPARQL

SPARQL adalah sebuah *RDF Query Language* dan akses data protokol untuk *Semantic Web*. *Query SPARQL* ini dapat terdiri dari *triple patterns*, konjungsi (*or*), dan disjungsi (*and*).

Rekomendasi *W3C* terhadap *SPARQL* terdiri dari tiga spesifikasi yang terpisah. Yang pertama spesifikasi *SPARQL Query Language* yang membentuk inti. Bersama dengan spesifikasi bahasa ini adalah spesifikasi *SPARQL Query XML Result Format* yang mendeskripsikan format *XML* untuk serialisasi hasil *query SPARQL* (termasuk *query SELECT* dan *ASK*). Spesifikasi yang ketiga adalah spesifikasi *SPARQL Protocol for RDF* untuk mendefinisikan *HTTP* sederhana *query database RDF*.

Beberapa *tool* dan *API* yang dapat digunakan untuk menjalankan *SPARQL* adalah *ARQ*, *Rasqal*, *RDF query*, *twingql*, *Pellet*, dan *KAON2*. (Leigh, 2005)

```

PREFIX obp: <http://lib.uin-malang.ac.id/ontologies/Library.owl#>
PREFIX dtp: <http://lib.uin-malang.ac.id/ontologies/Library.owl#>
SELECT ?l1 ?t1 ?s1 WHERE {
?l2 obp:advisor1Of ?s2. ?l2 dtp:name ?l1. ?s2 dtp:name ?s1.
?s2 dtp:write ?t2. ?t2 dtp:title ?t1 }ORDER BY ?l1

```

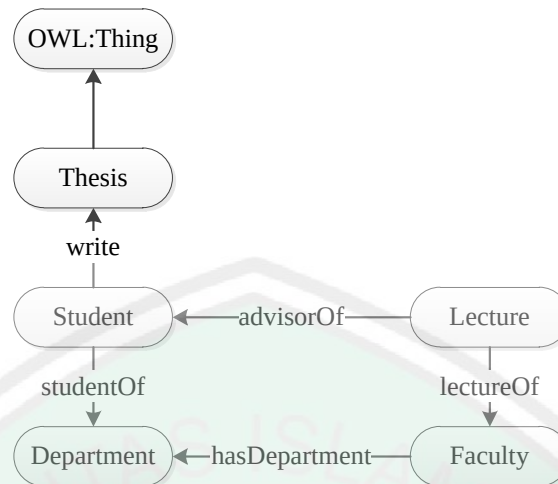
Gambar 2.5 Contoh SPARQL Query (Umar Faruq, 2015)

Menurut Lintang (2012), pengertian lain *SPARQL* adalah *query* untuk *RDF/OWL*, *query* ini digunakan untuk mengambil data yang ditulis dengan menggunakan *RDF/OWL* atau *XML*. Contoh *SPARQL Query* ditunjukkan pada **Gambar 2.5**.

2.10 *Ontology*

Awal mula *ontology* berasal dari ilmu filsafat dan menjadi penelitian populer di bidang *computer science* dan *information system*. Menurut *Aristoteles*, *ontology* adalah “*study of existence*”, studi tentang keberadaan, yakni suatu sistem kategori atau sistem klarifikasi untuk menjelaskan keberadaan dunia nyata. *Aristoteles* tidak pernah menyebutkan kata “*ontology*” seperti yang diperkenalkan oleh *Rudolf Gockel* dalam *Lexicon Philosophicum* dan *Jacob Lorhard* dalam *Ogdaes Scholasticas*. *Aristoteles* meletakkan dasar-dasar untuk klasifikasi *ontology*, dan mengembangkan sistem untuk klasifikasi.

Ontology merupakan model dari dunia yang direpresentasikan dengan *tree* yang tidak beraturan yang terbentuk dari sekumpulan konsep (*concept*) yang saling terhubung. Konsep adalah entitas abstrak yang tidak bergantung pada bahasa (bersifat *language-independent*) dan bukan kata-kata. Konsep-konsep tersebut diekspresikan dalam *ontology* dengan menggunakan kata/frase bahasa Inggris atau bahasa lain hanya sebagai konvensi untuk penyederhanaan.



Gambar 2.6 Contoh *Ontology* (Umar Faruq, 2015)

Mesin atau computer tidak akan memahami secara langsung maksud dari konsep tersebut, sehingga sekalipun suatu konsep direferensikan dengan angka, simbol, atau kode maka hal tersebut tidak akan menjadi masalah. Tujuan dibuatnya *ontology semantic* adalah untuk meningkatkan otomatisasi pemrosesan teks dengan menyediakan representasi konsep yang ada di dunia secara *language independent* dan *meaning-based*. Contoh *ontology* ditunjukkan pada **Gambar 2.6**.

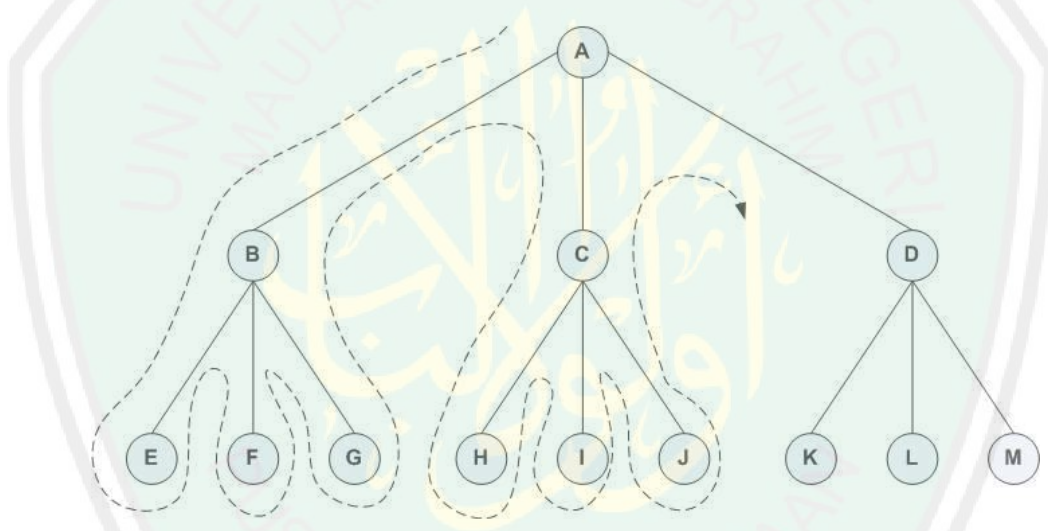
2.11 Algoritma *Depth First Search*

Depth first search (DFS) adalah algoritma umum traversal atau pencarian pada pohon, struktur pohon, atau graf.

Depth first search adalah penelusuran graph yang arah penelusurannya mendahulukan ke arah kedalaman graf tersebut (Sjukani, 2007). Algoritma pencarian mendalam *Depth-First Search (DFS)* adalah proses searching sistematis buta yang melakukan ekspansi sebuah path (jalur) menuju penyelesaian masalah sebelum melakukan eksplorasi terhadap path yang lain. Proses searching mengikuti sebuah path tunggal sampai menemukan goal atau dead end. Apabila proses

searching menemukan dead-end, DFS akan melakukan penelusuran balik ke node terakhir untuk melihat apakah node tersebut memiliki path cabang yang belum dieksplorasi. Apabila cabang ditemukan, DFS akan melakukan cabang tersebut. Apabila sudah tidak ada lagi cabang yang dapat dieksplorasi, DFS akan kembali ke node parent dan melakukan proses searching terhadap cabang yang belum dieksplorasi dari node parent sampai menemukan penyelesaian masalah. Urutan proses searching DFS ditunjukkan dalam **Gambar 2.7** adalah: A, B, E,F, G, C, ...

Figure



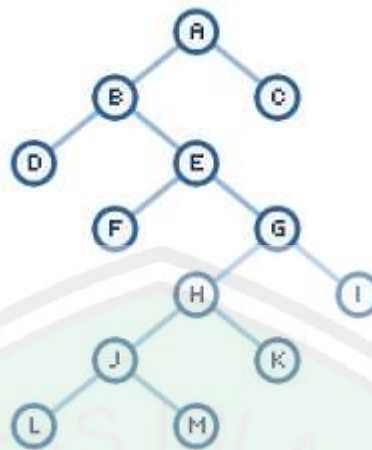
Gambar 2.7 Diagram pohon dari DFS (Sjukani, 2007)

Pencarian dilakukan pada satu node dalam setiap level dari yang paling kiri. Jika pada level yang paling dalam, solusi belum ditemukan, maka pencarian dilanjutkan pada node sebelah kanan. Node yang kiri dapat dihapus dari memori. Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada level sebelumnya. Demikian seterusnya sampai ditemukan solusi. Jika solusi ditemukan maka tidak diperlukan proses backtracking (penelusuran balik untuk mendapatkan jalur yang diinginkan).

Depth First Search adalah pencarian yang berjalan dengan meluaskan anak akar pertama dari pohon pencarian yang dipilih dan berjalan dalam dan lebih dalam lagi sampai simpul tujuan ditemukan, atau sampai menemukan simpul yang tidak punya anak. Kemudian, pencarian backtracking, akan kembali ke simpul yang belum selesai ditelusuri. Dalam implementasi nonrekursif, semua simpul yang diluaskan ditambahkan ke LIFO stack untuk penelusuran.

Beberapa hal mengenai traversal DFS :

- a. Traversal DFS dari suatu graf G akan :
 - Mengunjungi semua simpul dan sisi dari G
 - Menentukan apakah G terhubung
 - Memperhitungkan komponen terhubung dari G
 - Memperhitungkan pohon merentang dari G
- b. DFS pada graf dengan n simpul dan m sisi membutuhkan waktu $O(n + m)$
- c. DFS dapat lebih jauh diperluas untuk menyelesaikan masalah-masalah graf
 - Menemukan dan melaporkan lintasan antara dua simpul yang diberikan.
 - Menemukan sirkuit pada graf
- d. Depth First Search untuk menggambar graf Euler ke pohon biner terdapat suatu pohon pencarian.



Gambar 2.8 Pohon pencarian (search tree) (Hafid Inggiantowi, 2008)

Pada *graf* yang kita miliki **Gambar 2.8**, hubungan antara lintasan tidak dua arah. Semua lintasan hanya berjalan dari atas ke bawah. Dengan kata lain, A memiliki lintasan ke B dan C, tapi B dan C tidak memiliki lintasan ke A. Jadi, secara umum bisa kita anggap seperti jalan satu arah (one way street).

Setiap lingkaran yang berhuruf pada graf di atas adalah simpul. Setiap simpul dapat dihubungkan dengan simpul yang lain dengan sisi atau suatu lintasan, dan simpul yang langsung berhubungan dengan simpul disebut tetangga. B dan C adalah tetangga dari A. Kemudian, E dan D adalah tetangga dari B, dan B bukan tetangga dari D atau E karena B tidak bisa dicapai menggunakan D atau E.

Berikut ini akan dijelaskan demonstrasi yang dapat membantu kita memahami cara kerja *Depth First Search*. Misalkan kita ingin mencari lintasan antara simpul A dan F pada graf yang telah kita punyai diatas (**Gambar 2.8**)

Depth First Search telah kita ketahui, bekerja dengan memilih suatu simpul, kemudian akan memeriksa tetangga-tetangganya, meluaskan dengan simpul pertama yang ditemui pada tetangga-tetangganya, kemudian memeriksa apakah

simpul hasil perluasan adalah tujuannya atau bukan, dan jika bukan, akan melanjutkan menelusuri simpul yang lebih banyak lagi.

Langkah Awal:

Dimulai dari simpul akar, atau simpul tujuan yang kita pilih, yaitu A



Gambar 2.9 Langkah Awal (Hafid Inggiantowi, 2008)

Kita akan menggunakan dua list info untuk menyimpan langkah yang kita lakukan, yaitu Open dan Close. Open akan menyimpan langkah yang perlu kita lakukan, dan Close akan menyimpan langkah yang telah kita lakukan. Saat ini, kita hanya punya titik awal yang kita pilih, yaitu simpul A. Kita belum melakukan apapun ke simpul tersebut, sehingga sekarang kita tambahkan ke Open.

Open : A

Close : < kosong >

Langkah 1

Sekarang, kita memeriksa tetangga- tetangga dari simpul A. Dengan kata lain, mengambil simpul awal dari Open kita dan kemudian menelusuri tetangga- tetangganya.



Gambar 2.10 Langkah 1 (Hafid Inggiantowi, 2008)

Tetangga-tetangga dari simpul A adalah simpul B dan C. Karena sekarang kita telah selesai dengan simpul A, kita dapat menghapusnya dari Open dan menambahkannya ke Close. Kita belum selesai dengan langkah ini tentunya. Tambahkan dua simpul B dan C tadi ke Open.

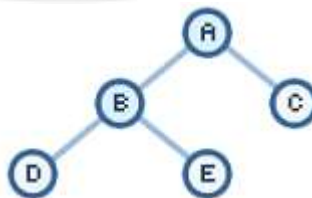
Sekarang, Open dan Close berisi data sebagai berikut :

Open : B, C

Close: A

Langkah 2

Open kini berisi dua simpul. Untuk depth first search dan breadth first search, selalu kita telusuri simpul pertama dari Open. Simpul pertama pada open sekarang adalah simpul B. B bukan tujuan kita, sehingga kita telusuri tetangga-tetangganya.



Gambar 2.11 Langkah 2 (Hafid Inggiantowi, 2008)

Karena kini B telah diluaskan, kita dapat menghapusnya dari Open dan menambahkannya ke Close. Kini, simpul baru kita adalah D dan E, dan kita tambahkan simpul ini ke awal dari Open (menggantikan B yang telah dihapus). Open: D, E, C

Close: A, B

Langkah 3 :

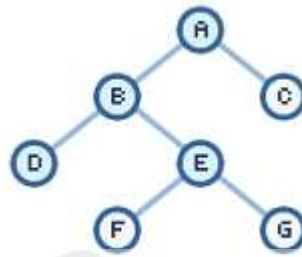
Kita akan melihat bahwa sebuah pola mulai terbentuk. Karena D ada di awal dari Open, maka kita meluaskannya. D bukan tujuan kita, dan tidak berisi satu pun tetangga. Selanjutnya, kita dapat menghapus D dari Open dan menambahkannya ke Close.

Open: E, C

Close: A, B, D

Langkah 4:

Sekarang kita meluaskan simpul E dari Open. E bukan tujuan kita, sehingga kita telusuri tetangganya dan menemukan bahwa ia memiliki tetangga F dan G. Karena F adalah tujuan kita, kita masih belum berhenti di sini tentunya. Di samping F berada pada lintasan kita, kita hanya akan berhenti begitu kita akan meluaskan simpul tujuan kita, dalam kasus ini yaitu F:



Gambar 2.11 Langkah 4 (Hafid Inggiantowi, 2008)

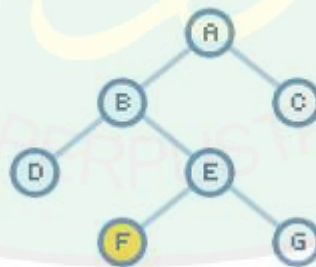
Kita akan menghapus simpul E dari Open dan menambahkan simpul F dan G ke dalamnya. Kemudian, simpul E ditambahkan ke Close:

Open: F, G, C

Close: A, B, D, E

Langkah 5

Sekarang kita meluaskan simpul F. Karena F adalah tujuan kita, maka kita berhenti



Gambar 2.12 Langkah 5 (Hafid Inggiantowi, 2008)

Kita menghapus F dari Open dan menambahkannya ke Close. Karena kita telah sampai di tujuan, maka tidak perlu lagi meluaskan F dan mencari tetangganya. Open dan Close akhir kita akan berisi data sebagai berikut :

Open: G, C

Close: A, B, D, E, F

Lintasan akhir yang akan diambil oleh metode *Depth First Search* kita adalah nilai terakhir dari Close, yaitu : A, B, D, E, F.



BAB III

METODE PENELITIAN

3.1 Objek Penelitian

Objek dalam penelitian ini yaitu perancangan *resource description framework* untuk pembuatan *knowledge base digital library* Universitas Islam Negeri Maulana Malik Ibrahim Malang.

3.2 Alat Penelitian

Alat penelitian berupa *hardware* dan *software* dijelaskan pada subbab selanjutnya.

3.2.1 Hardware yang Digunakan

Perangkat keras yang digunakan dalam penelitian ini adalah dengan spesifikasi sebagai berikut:

- a. Intel Pentium B960 Sandy Bridge @ 2.20 Ghz
- b. RAM 3.00 Gigabyte
- c. Hardisk 500 Gigabyte
- d. VGA Intel HD Graphics Family
- e. Keyboard dan Mouse
- f. Koneksi Internet (Wifi)

3.2.2 Software yang Digunakan

Adapun perangkat lunak yang digunakan dalam penelitian ini adalah:

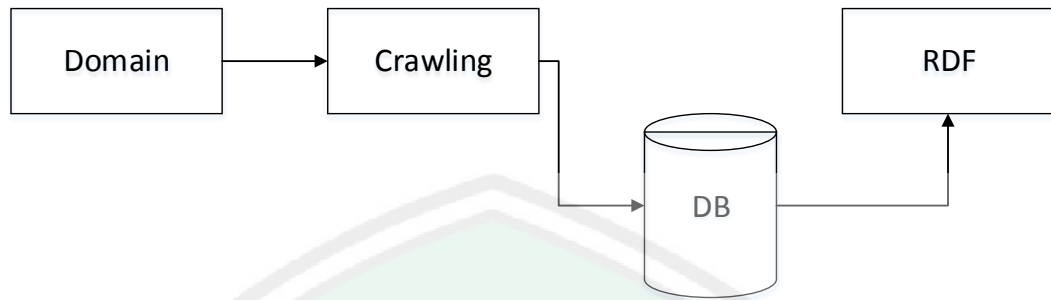
- a. Operating System Windows 8.1 Pro
- b. Java Runtime Environment 8 Build 40

- c. *Notepad++* 6.7.5
- d. *XAMPP* 2.3.1
- e. *Jena Fuseki* 1.1.1
- f. *Neatben* 8.0
- g. *Mozilla Firefox* 37.0

3.3 Deskripsi Sistem

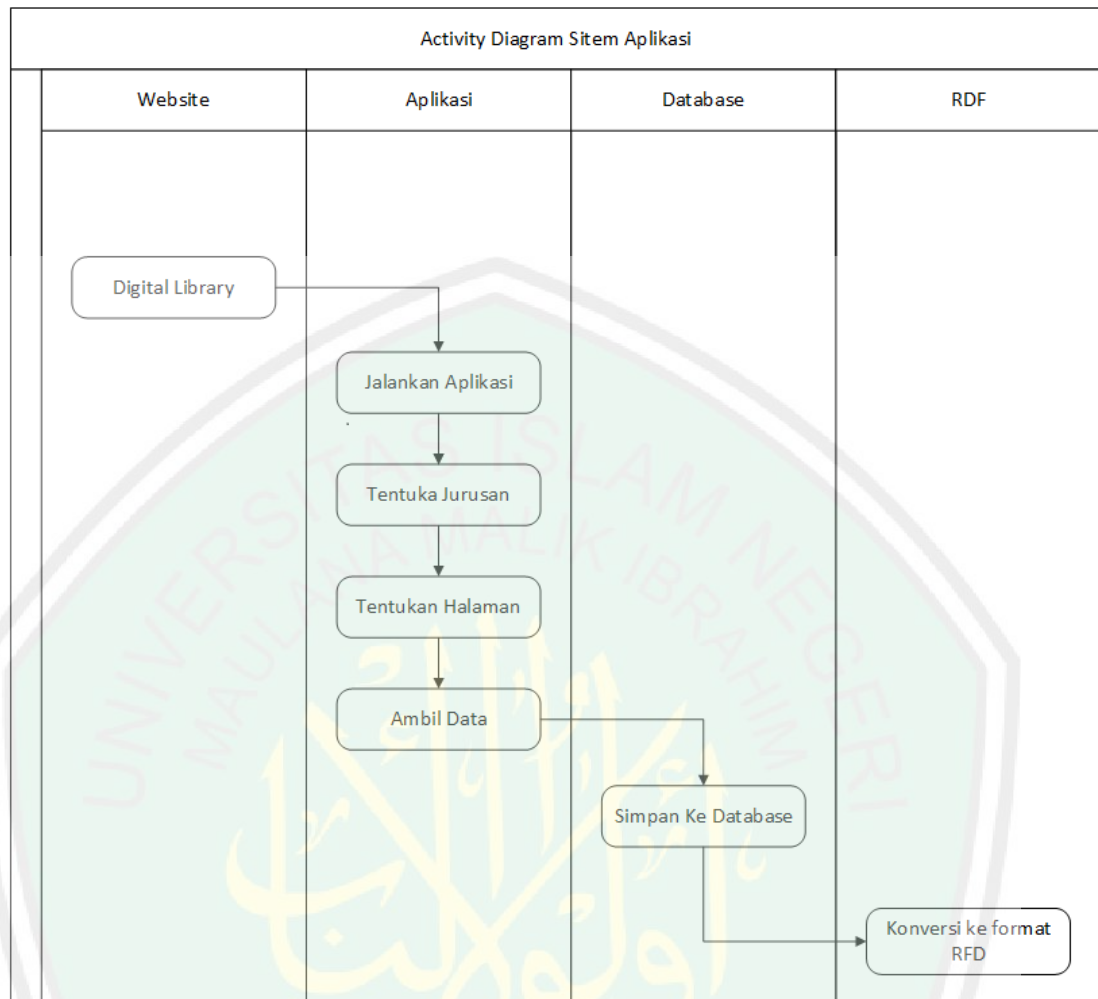
Perancangan merupakan tahapan yang penting dalam pembuatan suatu sistem, sehingga dengan perancangan yang baik diharapkan akan dihasilkan suatu sistem yang sesuai dengan perancangan yang baik diharapkan akan dihasilkan suatu sistem yang sesuai dengan fungsi dan tujuan dari dibuatnya sistem tersebut.

Perancangan pada sistem ini utamanya terbagi menjadi dua, yaitu perancangan *Ontology* sesuai domain *digital library* dengan menggunakan aplikasi ekstraksi dan perancangan *resource description framework* untuk pembuatan *knowledge base digital library*. Pada perancangan *Ontology* mengenai bagaimana pembuatan *Ontology*, yaitu mulai penentuan domain sesuai kaidah serta objek dan ruang lingkupnya hingga bagaimana membangun aplikasi ekstraksi yang berguna untuk mempermudah pembuatan *Ontology* yang sesuai dengan skema *Ontology* pada *digital library*. Sedangkan pada perancangan *resource description framework* menjelaskan mengenai perancangan *resource description framework* dengan menggunakan hasil output data dari proses ekstraksi sebelumnya. Untuk lebih jelasnya bisa dilihat pada blok diagram dan pada **Gambar 3.1** dibawah ini.



Gambar 3.1 Blok Diagram

Sedangkan untuk proses activity diagram berisi secara detil lagi terkait desain sistem dari aplikasi ini, untuk lebih jelasnya bisa dilihat pada **Gambar 3.2** dibawah ini.



Gambar 3.2 Activity Diagram Aplikasi

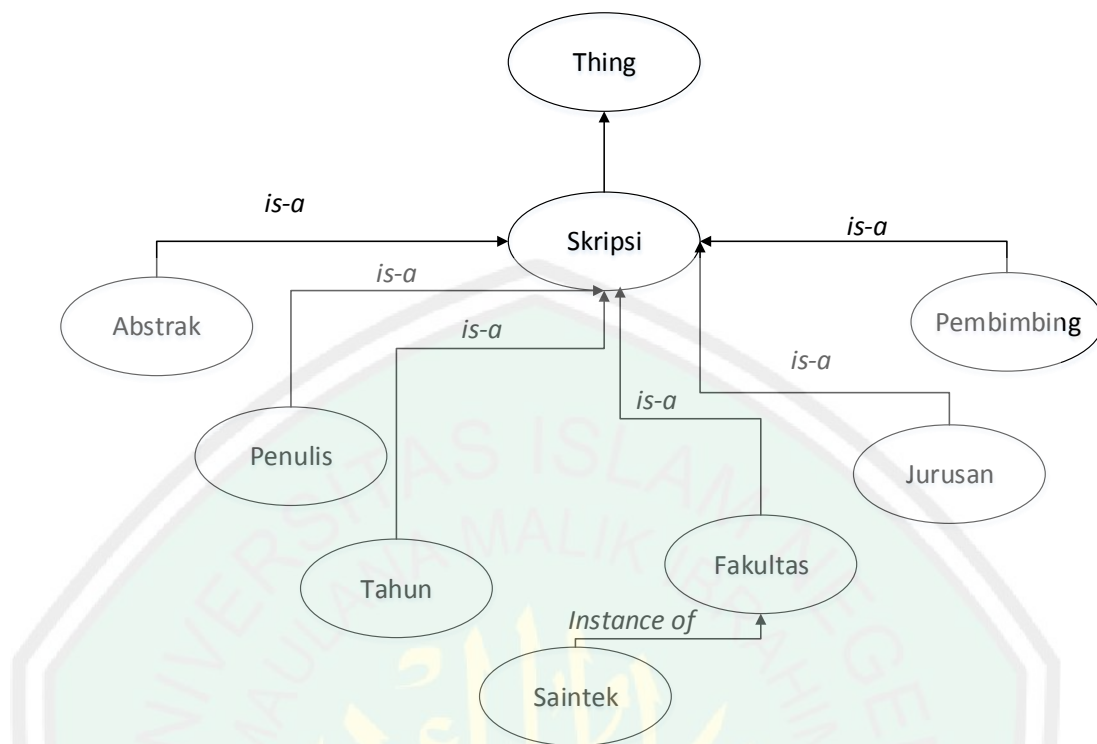
3.4 Pemodelan Data *Digital Library* Menggunakan *Ontology*

Sebelum melakukan proses ini hal pertama yang akan dipersiapkan adalah penentuan objek, domain, ataupun lainnya yang diperlukan untuk pembuatan yang sesuai dengan kaidah pembuatan *Ontology*.

Ontology merupakan bentuk dari suatu pengetahuan tentang dunia nyata yang kemudian direpresentasikan dalam suatu term yang terdiri dari sekumpulan konsep (concept) yang saling terhubung. Dari kumpulan konsep tersebut akan membentuk suatu tree yang menunjukkan adanya relasi antar konsep. Konsep

merupakan entitas abstrak yang diekspresikan dalam *Ontology* dengan menggunakan istilah tertentu sebagai konvensi penyederhanaan. Hal tersebut bertujuan agar nantinya mesin atau komputer dapat memahami maksud dari konsep tersebut. *Ontology* dilakukan untuk meningkatkan otomatisasi pemrosesan teks dengan menyediakan representasi konsep yang telah dibuat. Makna dari *term* yang telah dibuat itulah yang nantinya akan diinterpretasikan oleh mesin dan manusia. *Ontology* hanya mencakup *term* yang memuat konsep paling umum atau konsep utama (*super concept*) yang menarik.

Bagi pemahaman manusia, konsep utama merupakan suatu *class* yang selanjutnya akan diidentifikasi dan diimplementasikan dalam *term* yang lebih spesifik kedalam bentuk *subclass* (ditandai dengan *term is-a*) yang saling berhubungan. Relasi tersebut menunjukkan bahwa suatu *class* (*super concept*) adalah lebih umum daripada yang lain (*subclass*). *Class* yang umum berada diatas *class* yang lebih spesifik dan pada umumnya *class* yang berada paling atas disebut dengan istilah “Thing”. Sebagai contoh, *class* Skripsi lebih umum daripada *class* Fakultas. Pada **Gambar 3.3** menunjukkan contoh hirarki is-a dimana *class* yang lebih umum berada di atas *class* yang lebih spesifik



Gambar 3.3 Hirarki is-a

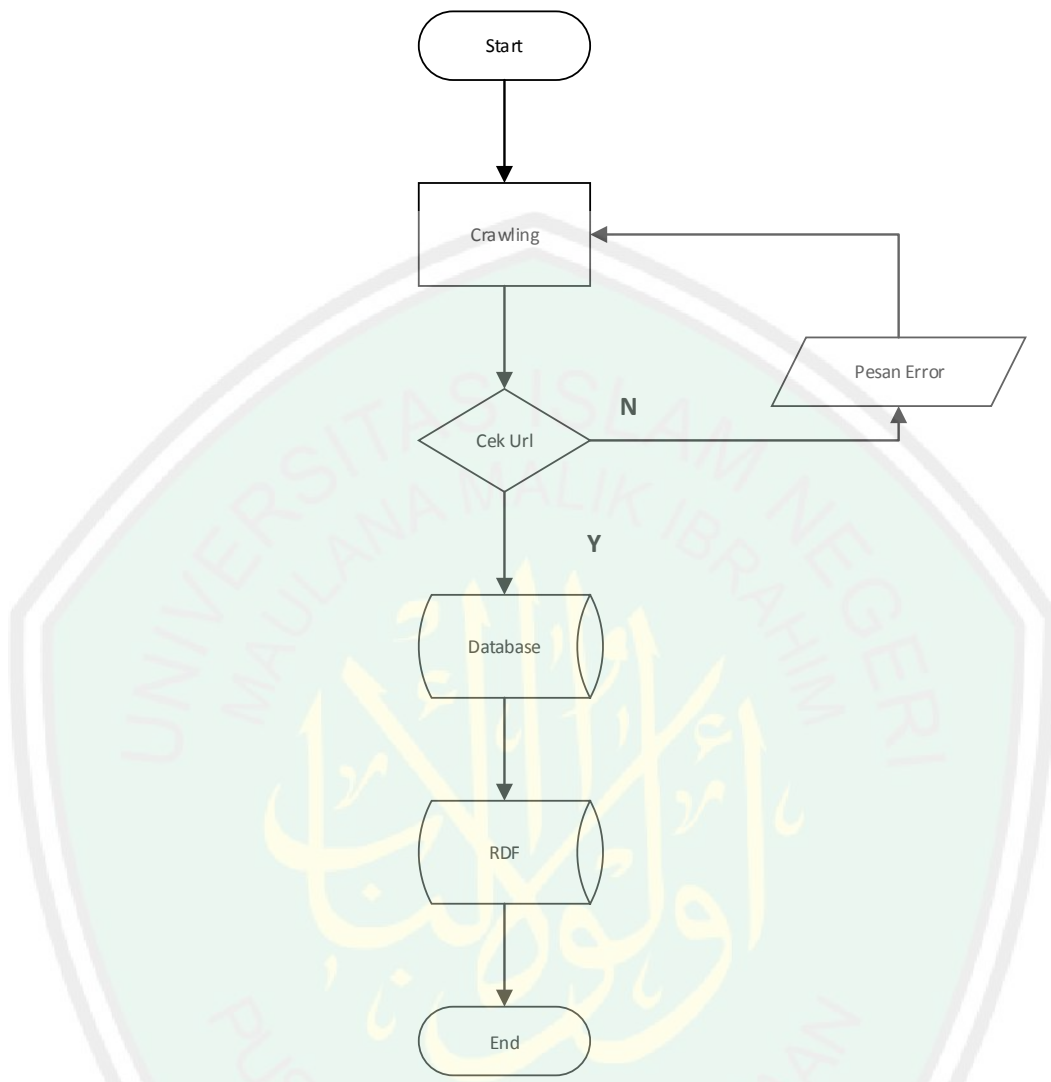
Gambar hirarki pada **Gambar 3.3** dapat memberikan pemahaman apabila *class* Skripsi lebih umum dari *class* Abstrak, Penulis, Tahun, Fakultas, Jurusan maupun Pembimbing. Hal tersebut menjelaskan bahwa setiap Fakultas (*subclass*) adalah bagian dari Skripsi (*class*). *Class* menjelaskan satu set objek. Sebagai contoh, Saintek merupakan salah satu bagian (*instance*) dari Fakultas. Sehingga adanya *class* Fakultas bertujuan untuk meng-capture semua Fakultas yang ada. Relasi pada *Instance of* mempunyai arti sebagai object aktual yang di capture oleh *class* Fakultas. Sehingga dalam relasi is-a antara *class* Fakultas dan *class* Skripsi akan menyatakan secara eksplisit apabila Saintek harus menjadi *instance* dari *class* Fakultas dan *class* Skripsi. 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. Dalam ini, apa yang dimengerti oleh mesin masih tidak sejauh apa yang dimengerti oleh manusia, namun dengan pengetahuan yang dapat diketahui oleh manusia direpresentasikan dan dikodekan sehingga mesin dapat memprosesnya dan dapat menyimpulkan pengetahuan tersebut.

3.5 Membangun Aplikasi *Digilib Extractor*

Web extractor adalah salah satu bagian terpenting pada sistem ini. Dimana *Web Extractor* berfungsi untuk mengambil data sesuai dari *Web domain* yang telah ditentukan sebelumnya.

Aplikasi ini akan merancang dan membuat sebuah *data resource description framework* dengan beberapa tahapan proses dari berbagai format atau struktur metadata yang ada di *digital library* dari semula yang tidak terstruktur menjadi lebih terstruktur sehingga aplikasi ini berfungsi untuk sebuah merancang dan membuat data berupa *resource description framework*. Sedangkan untuk tahapan prosesnya yang terdapat pada aplikasi ini terdiri dari : pengumpulan data ontology yang ada di dalam metadata (crawling), poses pencarian data metadata dari direktori *digital library* (searching), proses penginputan data ke dalam database kemudian proses terakhir yaitu pembuatan RDF. Untuk lebih jelasnya dapat dilihat pada **Gambar 3.4** Tentang flowchart dari aplikasi ini.



Gambar 3.4 Desain Flowchart Aplikasi

Crawler merupakan program yang berjalan secara otomatis, berisi script program yang melakukan crawling melalui halaman website mengumpulkan data berdasarkan indeks dari halaman web yang ditentukan. Nama alternatif untuk crawler seperti spider, robot dan automatic indexer. Sebuah crawler dapat untuk berbagai tujuan. Penggunaan paling umum terkait dengan mesin pencari.

Mesin pencari menggunakan 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.

Dalam penelitian ini crawler digunakan untuk mengumpulkan metadata yang diperoleh dari sebuah alamat website di www.lib.uin-malang.ac.id. Ada sub proses dalam proses pengumpulan data dengan menggunakan crawler ini. Dimulai dari menentukan alamat web yang akan diambil datanya, kemudian membuat koneksi terhadap alamat yang telah ditentukan sebelumnya, apabila koneksi gagal akan diberikan informasi bahwa koneksi tidak dapat dilakukan, apabila koneksi berhasil maka akan dilanjutkan proses selanjutnya yaitu mendownload informasi yang ada di alamat web tersebut.

Untuk proses penelusuran halaman skripsi *digital library*, aplikasi ini menggunakan kaidah atau konsep dari algoritma *Depth-first search* (DFS). *Depth-first search* (DFS) melakukan pencarian secara preorder. Mengunjungi anak suatu simpul sebelum simpul tetangganya. Berkaitan dengan mesin pencari, DFS ini cenderung mengindeks dokumen berdasarkan suatu link yang ada pada *digital library*.

Selanjutnya adalah proses pengunduhan akan dilakukan sampai seluruh informasi yang ada di alamat website tersebut sudah tidak ada lagi yang diambil, termasuk link-link indeks yang terhubung dengan alamat web tersebut. Pengunduhan informasi dapat berupa data teks seperti Nim, Judul

Skripsi, Nama Penulis, Dosen Pembimbing, Tahun, dan sebagainya. Hasil pengunduhan akan disimpan secara otomatis ke dalam database.

Pada langkah selanjutnya terkait dari bagaimana menyimpan hasil unduhan ke dalam database, Sehingga sistem ini berfungsi dengan semestinya dimulai dari memasukkan data dokumen metadata yang telah diunduh melalui crawler ke dalam tabel database. Tahapan prosesnya yaitu menjalankan modul pendataan file teks, dimana modul ini akan digunakan untuk melakukan pencarian dokumen teks dan sesuai dengan kriteria. Dalam penelitian ini kriteria yang dimaksud pencarian dokumen berupa metadata pada *digital library*.

Pencarian dilakukan dengan menentukan direktori termasuk subdirektorinya tersebut berada. Ketika dokumen tersebut ditemukan maka hasilnya akan ditampilkan berserta nama subdirektorinya. Selanjutnya dilakukan proses memasukkan hasil pencarian dokumen yang dicari ke dalam tabel database sekaligus pembuatan RDF. Saat melakukan proses tersebut ditampilkan isi dokumen, seperti id, Nim, Judul Skripsi, Nama Penulis, Dosen Pembimbing, Tahun, dan sebagainya.

Proses memasukkan dokumen metadata ke tabel database dan pembuatan RDF akan membutuhkan beberapa waktu tergantung dari banyaknya hasil pencarian yang dilakukan sebelumnya. Apabila hasil pencariannya banyak maka waktu yang dibutuhkan akan lebih lama, sebaliknya apabila hasil pencarian dokumen yang dilakukan menghasilkan jumlah file relatif sedikit maka waktu yang dibutuhkan lebih pendek. Disamping itu faktor kecepatan koneksi internet juga mempengaruhi.

3.6 Perancangan Database

Desain database untuk aplikasi ini menggunakan satu database. Penelitian ini menggunakan database *SQL* karena memerlukan data yang jumlahnya tidak sedikit. Ada satu tabel yang digunakan pada database ini yaitu tabel hasil, untuk lebih jelasnya bisa dilihat dari **Gambar 3. 5**

Tabel hasil ini digunakan untuk menyimpan data semua data hasil dari proses ekstraksi yang disesuaikan dengan perancangan skema *ontology digital library* seperti id mahasiswa yang berisi Nomor Induk Mahasiswa (NIM), id jurusan yang berisi id masing-masing jurusan, Link data, link data ini berisi alamat dari halaman skripsi dari masing-masing id mahasiswa, judul pada id ini berisi tentang judul skripsi dari tiap-tiap mahasiswa, penulis pada id ini berisi tentang nama mahasiswa selaku penulis skripsi, tahun pada id ini berisi tentang tahun pembuatan skripsi oleh tiap-tiap mahasiswa, fakultas pada id ini berisi informasi tentang fakultas dari masing-masing penulis, jurusan pada id ini berisi tentang informasi jurusan dari masing-masing penulis, pembimbing pada id ini berisi informasi tentang nama dosen pembimbing dari masing-masing penulis, keyword pada id ini berisi tentang informasi kata kunci dari abstrak masing-masing judul skripsi, dan id isi berisi tentang isi dari abstrak dari masing-masing judul skripsi.



db_digilib.tb_hasil	
ID	int(11)
ID_jur	varchar(255)
Link_Data	varchar(255)
Judul	varchar(255)
Penulis	varchar(255)
Tahun	varchar(4)
Fak	varchar(255)
Jurusan	varchar(255)
Pembimbing	varchar(255)
Keyword	varchar(255)
Isi	text
Data_Html	text

Gambar 3.5 Desain Tabel *Digital Library*

3.7 Perancangan *Resource Description Framework (RDF)*

Resource Description Framework (RDF) adalah general purpose language untuk merepresentasikan informasi dalam web. RDF memberikan interoperability antar aplikasi dalam mempertukarkan informasi yang machine-understandable di web. RDF memfasilitasi pemrosesan ter-otomatisasi terhadap web resource. RDF dapat digunakan dalam berbagai aplikasi, sebagai contoh: dalam resource discovery untuk memberikan kemampuan lebih baik pada search engine, dalam katalog untuk menjelaskan tentang konten dan hubungan antar konten.

World Wide Web Consortium (W3C) mengenalkan RDF (*resource description framework*) untuk menjadikan metadata standar yang dapat dimengerti oleh antar mesin untuk mendeskripsikan resource yang ada diweb. RDF dapat dilihat sebagai suatu resource dalam suatu format yang dapat dimengerti oleh semuanya.

Seperti layaknya HTML, RDF/XML memiliki sifat mudah diproses oleh mesin dan, dengan URI, dapat menghubungkan beberapa informasi lintas web. Namun, berbeda dengan hypertext yang konvensional, RDF URI dapat merujuk kepada segala sesuatu selama dapat diidentifikasi, termasuk sesuatu yang tidak dapat diperoleh secara langsung dari web (seperti Person : Eric Miller).

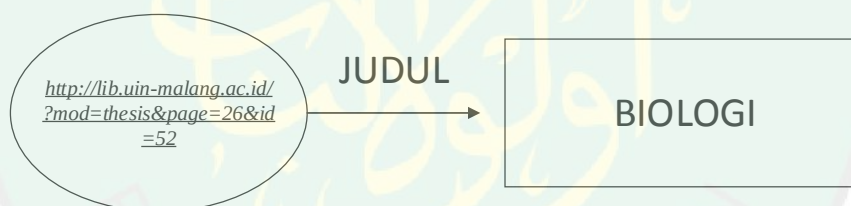
Singkat kata RDF disamping dapat digunakan untuk sesuatu seperti halaman web, RDF juga dapat mendeskripsikan mobil, bisnis, orang, berita, event dsb. Selain itu RDF properties juga memiliki URI, untuk mengidentifikasi secara akurat hubungan / relationship antara item-item yang berhubungan. Sintaks RDF, model data RDF menyediakan suatu kerangka konseptual untuk mendefinisikan dan menggunakan metadata. Untuk menciptakan dan mempertukarkan metadata dibutuhkan suatu sintaks yang konkret. Manola dan Miller (2004) mengemukakan spesifikasi RDF ini biasanya menggunakan sintaks dasar XML sehingga disebut juga dengan RDF/XML.

RDF (*Resource Description Framework*) merupakan metadata yang digunakan untuk mendeskripsikan alamat sumber daya pada web. Metadata ini dapat berupa judul, pengarang, hak cipta, dan lisensi dalam dokumen web. Elemen pernyataan dalam RDF terdiri dari subjek, predikat dan objek. Subjek adalah sesuatu yang dideskripsikan dan biasanya berisikan alamat URI.

Dalam hal ini alamat URI merepresentasikan sumber daya. Predikat merupakan properti dari sumber daya yang menjelaskan hubungan antara subjek dengan objek. Selain itu objek merupakan nilai dari sebuah predikat. Objek mempunyai dua tipe data yaitu objek yang mempunyai tipe URI misalnya

“<http://lib.uin-malang.ac.id/?mod=thesis&page=26&id=52>” dan objek yang bertipe literal misalnya “biologi”. Subjek dan predikat berisikan data yang bertipe sumber daya sedangkan objek dapat bertipe sumber daya dan literal.

Dalam RDF, jika kita ingin mengekspresikan kalimat Judul berita pada link <http://lib.uin-malang.ac.id/?mod=thesis&page=26&id=52> adalah biologi maka untuk menyesuaikan dengan struktur RDF, kalimat bisa kita ubah menjadi <http://lib.uin-malang.ac.id/?mod=thesis&page=26&id=52> mempunyai judul skripsi biologi. Dari kalimat tersebut kita dapatkan <http://lib.uin-malang.ac.id/?mod=thesis&page=26&id=52> sebagai subjek, “judul” sebagai predikat dan “biologi” sebagai objeknya. Berikut representasi pernyataan dalam bentuk grafik:



Bentuk elipse digambarkan sebagai subjek, panah sebagai predikat dan kotak sebagai objek. Contoh implementasi RDF dalam bentuk XML adalah pada

Gambar 3.6

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skripsi="http://lib.uin-
malang.ac.id/skripsi#"><rdf:Description rdf:about="http://lib.uin-
malang.ac.id/?mod=th_detail%26id=04520006">
  <skripsi:judul>Biologi.</skripsi:judul>
</rdf:Description>
</rdf:RDF>
```

Gambar 3.6 Contoh Implementasi RDF

Pada **Gambar 3.6**, RDF menggunakan prefiks `rdf` dan skripsi untuk menyingkat alamat penuh URI. Kata `rdf` merupakan nama namespace yang mengacu pada syntax RDF sedangkan berita skripsi elemen elemen judul. Kedua prefiks tersebut dinamakan juga dengan namespace. Dalam namespace, walaupun kita merujuk ke alamat URI sebuah schema, kenyataannya alamat tersebut tidak harus selalu ada. Tidak ada aktifitas validasi dilakukan terhadap alamat tersebut. Hal ini dilakukan lebih untuk menghindari terjadinya penggunaan elemen yang ambigu. Caranya yaitu dengan membuat prefiks yang yang berlainan. Prefiks namespace merupakan bagian dari XML qualified name (Qname) yang digunakan untuk memudahkan penulisan RDF. Qname terdiri dari prefiks dan nama lokal. Misalnya teks `berita:judul`, kata `berita` merupakan nama pendek dari URI <http://lib.uin-malang.ac.id/?mod=thesis&page=26&id=52> sedangkan kata `judul` adalah elemen dari berita seperti pada link http://lib.uin-malang.ac.id/?mod=th_detail&id=08620003.

Sedangkan untuk perancangan RDF pada objek penelitian ini berdasarkan pada web <http://lib.uin-malang.ac.id/>, dimana jenis halaman web ini merupakan informasi skripsi dari berbagai jurusan maupun fakultas yang berada dilingkungan kampus Universitas Islam Negeri Maulana Malik Ibrahim Malang, pada halaman web <http://lib.uin-malang.ac.id/> terdapat beberapa informasi berupa nama penulis, Judul Skripsi, Tahun, Dosen Pembimbing, Keyword.

Untuk pendefinisain bentuk Subjek Predikat Objek (SPO) dapat direpresentasikan dari berbagai berbagai sudut informasi, contoh untuk Subjeknya adalah berdasarkan link atau alamat Nomor Induk Mahasiswa (NIM) dari halaman digital library, sedangkan untuk predikatnya berdasarkan dari apa yang mau di

representasikan contoh nama penulis, jurusan, tahun atau judul skripsi dari penulis, sedangkan untuk objeknya berisi tentang informasi dari predikat yang di cari. untuk megekspresikan perancangan bentuk Resource Description Frameworknya bisa dilihat pada **Gambar 3.7**

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:skripsi="http://lib.uin-
malang.ac.id/skripsi#"><rdf:Description
rdf:about=".....">
  <skripsi:judul>.....</skripsi:judul>
  <skripsi:penulis>.....</skripsi:penulis>
  <skripsi:tahun>.....</skripsi:tahun>
  <skripsi:fakultas>.....</skripsi:fakultas>
  <skripsi:jurusan>.....</skripsi:jurusan>
  <skripsi:pembimbing>.....</skripsi:pembimbing>
  <skripsi:keyword>.....</skripsi:keyword>
</rdf:Description>
</rdf:RDF>
```

Gambar 3.7 Rancangan RDF Digital Library

Setelah proses pendefinisian RDF dari digital library sesuai dengan kaidah RDF maka hasilnya akan bentuk Subjek Predikat Objek (SPO) dari *digital library* akan seperti **Gambar 3.8**

Tabel 3.1 Hasil Rancangan RDF *digital library*

Number	Subject	Predicate	Object
1	http://lib.uin-malang.ac.id/skripsi/	http://lib.uin-malang.ac.id/skripsi#penulis	"....."
2	http://lib.uin-malang.ac.id/skripsi/	http://lib.uin-malang.ac.id/skripsi#tahun	"....."
3	http://lib.uin-malang.ac.id/skripsi/	http://lib.uin-malang.ac.id/skripsi#fakultas	"....."
4	http://lib.uin-malang.ac.id/skripsi/	http://lib.uin-malang.ac.id/skripsi#jurusan	"....."
5	http://lib.uin-malang.ac.id/skripsi/	http://lib.uin-malang.ac.id/skripsi#pembimbing	"....."

Kemudian untuk hasil skema graphnya bisa dilihat pada **Gambar 3.8**



Gambar 3.8 Skema Graph dari RDF Digital Library

3.8 Perancangan Aplikasi

Tahapan selanjutnya adalah membangun *interface* untuk proses crawling (ekstraksi) serta pembuatan RDF *digital library* UIN Malang sehingga *user* dapat menggunakan aplikasi ini dengan mudah. Sistem ini dikemas dalam bentuk *digilib extractor*..

3.8.1 Halaman Utama Aplikasi

Aplikasi mempunyai satu tampilan utama, dimana didalamnya ada beberapa fitur seperti konfigurasi port database, pemilihan jurusan, penentuan jumlah halaman yang akan diekstraksi, pengulangan kembali proses ekstraksi, serta fitur pembuatan RDF.

The screenshot shows the initial interface of the 'APLIKASI EKSTRAKSI' (Extraction Application). It features a 'File' menu icon (a square with an 'X') in the top left. The main title 'APLIKASI EKSTRAKSI' is centered at the top. Below the title, there is a form with a 'Jurusan' (Department) dropdown menu showing 'Jurusan'. Below this is a 'Halaman' (Page) input field. To the right of the input field are three buttons: 'Get Data', 'Baru' (New), and 'Export Ke RDF'. Below the form is a large rectangular area labeled 'INFORMASI HASIL EKSTRAKSI' (Extraction Results Information). At the bottom, there is a small box containing the text 'Informasi Jumlah Page (Halaman) dan Proses Crawling (Ekstraksi)' (Information on the number of pages and the crawling process).

Gambar 3.9 Desain Halaman Awal aplikasi

3.8.2 Konfigurasi Database

Pada halaman ini berfungsi sebagai konfigurasi untuk pengisian port database. Sehingga pengguna dimudahkan untuk menentukan nama angka port untuk databasenya. Selain itu pengguna juga bisa menentukan nama serta password databasenya. Untuk lebih jelasnya bisa dilihat pada gambar 3. Dibawah ini.

The screenshot shows the 'Pengaturan Database' (Database Configuration) interface. It is titled 'File' at the top. Below the title is a box labeled 'Pengaturan Database'. Inside this box, there are five labels with corresponding input fields: 'DB HOST', 'PORT', 'DB NAME', 'USER', and 'PASSWORD'.

Gambar 3.10 Desain Interface Konfigurasi Database

3.9 Perancangan Uji Coba

Perancangan ujicoba akan dilakukan pada aplikasi ini adalah terkait dengan hasil perancangan *resource description framework* yang telah dihasilkan dengan menggunakan aplikasi digilib extractor ini. Pengujian ini dilakukan menggunakan *sparql* berdasarkan hasil *resource description framework digital library* UIN Maliki Malang. Tujuan dengan adanya perancangan ujicoba adalah untuk mengetahui keakurasian atau kevalidasian dari hasil perancangan *resource description framework digital library*.

Data yang digunakan dalam pengujian adalah sampel data sebanyak 18 data objek yang telah di inputkan sesuai dengan data skripsi yang ada pada digital library UIN Maliki Malang.

BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini dibahas mengenai rangkaian uji coba dan evaluasi yang telah dilakukan. Uji coba bertujuan untuk mengetahui tingkat keakurasian dari aplikasi dalam pembuatan RDF. Sedangkan evaluasi dilakukan bertujuan untuk analisa hasil uji coba agar mendapatkan kesimpulan dan saran untuk pengembangan aplikasi selanjutnya.

4.1 Implementasi Interface

Pada implementasi *interface* akan dijelaskan mengenai komponen-komponen dari aplikasi *digilib extractor* yang diimplementasikan pada pemograman java . Berikut beberapa tampilan java yang terdapat dalam aplikasi.

4.1.1 Tampilan Halaman Utama

Implementasi pada halaman utama ini muncul ketika program pertama kali dijalankan. Halaman utama ini memuat form untuk konfigurasi port database, disamping itu pada form utama ini memiliki beberapa tambahan fitur didalam, seperti pemilihan jurusan, jumlah halaman yang akan diekstraksi serta proses export ke RDF. Untuk lebih jelasnya interface dari halaman utama ditunjukkan oleh **Gambar 4.1**.



Gambar 4.1 Halaman Utama

4.1.2 Halaman Pengaturan Koneksi Database

Halaman ini digunakan untuk menampilkan fitur konfigurasi koneksi dari database seperti port database, nama database, user serta password dari database yang akan disimpan. Tujuan dari penambahan fitur ini adalah agar pengguna lebih dimudahkan dalam pengaturan akan data database yang akan disimpan. Untuk lebih jelasnya interface dari konfigurasi koneksi database dapat dilihat pada Gambar 4.2.



Gambar 4.2 Pengaturan Koneksi Database

```

Connection con = null;
String url = "jdbc:mysql://localhost:3306/";
String db = "db_digilib";
String driver = "com.mysql.jdbc.Driver";
String user = "root";
String pass = "";
String[][] keyData=null;

public MyQConn(FrmSetting fsetting) {

    this.url =
"jdbc:mysql://" + fsetting.txt1.getText().trim() + ":" + fsetting.txt2.g
etText().trim() + "/";
    this.db = fsetting.txt3.getText().trim();
    this.driver = "com.mysql.jdbc.Driver";
    this.user = fsetting.txt4.getText().trim();
    this.pass = fsetting.txt5.getText().trim();
}

```

Gambar 4.3 Source Code Pengaturan Koneksi Database

4.1.3 Fitur Penentuan Jurusan

Fitur ini digunakan untuk menampilkan combobox daftar jurusan sekaligus informasi fakultas yang ada pada *digital library*, pada menu ini fitur dari aplikasi ini berupa combo box yang berisi semua jurusan yang ada di UIN Maliki Malang. Seperti Jurusan Teknik Informatika, Pendidikan Agama Islam, Psikologi, Akuntansi Dan lain sebagainya. Untuk lebih jelasnya bisa dilihat pada **Gambar**

4.4



Gambar 4.4 Halaman Skripsi Perfakultas

```
String[] datacbJurusan = new String[21];

datacbJurusan[0] = ("[Tarbiyah] Pendidikan Agama Islam");
datacbJurusan[1] = ("[Tarbiyah] Ilmu Pengetahuan Sosial");
datacbJurusan[7] = ("[Psikologi] Psikologi");
datacbJurusan[8] = ("[Saintek] Matematika");
datacbJurusan[9] = ("[Saintek] Biologi");
datacbJurusan[10] = ("[Saintek] Kimia");
datacbJurusan[11] = ("[Saintek] Fisika");
datacbJurusan[12] = ("[Saintek] Teknik Informatika");
datacbJurusan[13] = ("[Saintek] Teknik Arsitektur ");

Jurusan = new HashMap<>();
Jurusan.put(datacbJurusan[0], "11");
Jurusan.put(datacbJurusan[1], "12");
Jurusan.put(datacbJurusan[2], "13");
Jurusan.put(datacbJurusan[3], "21");
Jurusan.put(datacbJurusan[4], "31");
Jurusan.put(datacbJurusan[5], "32");
```

Gambar 4.5 Source Code Halaman Skripsi Perfakultas

4.1.4 Fitur Penentuan Jumlah Pengambilan Data

Pada fitur ini berisi sebuah textfiel yang berfungsi sebagai penentu jumlah untuk pengambilan sebuah data yang ada pada digital library UIN Maliki Malang, dimana untuk angka nominal awal adalah satu sebagai angka pertama pada halaman skripsi yang ada pada *digital library* kemudian memasukkan angka nominal pada textfield yang telah disediakan. Untuk lebih detilnya bisa dilihat pada **Gambar 4.6**.



Gambar 4.6 Fitur Penentuan Jumlah Pengambilan Data


```
private void btnGetDataActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_btnGetDataActionPerformed
    if (txtFrom.getText().trim().equals(txTo.getText().trim())
|| txtFrom.getText().trim().isEmpty() ||
txTo.getText().trim().isEmpty()) {
        JOptionPane.showMessageDialog(rootPane, "Isikan nomer
halaman dengan bernar");
        return;
    }
    txtLog.setText("");
    Task progress = new Task("");
    progress.execute();
}
```

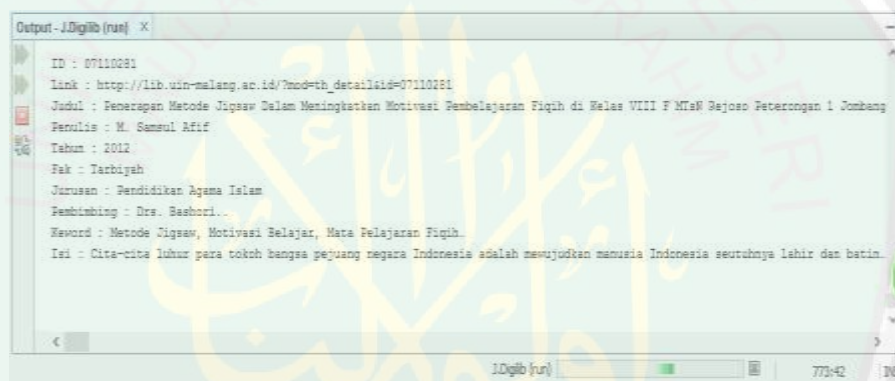
Gambar 4.7 Source Code Fitur Penentuan Jumlah Pengambilan Data

4.1.5 Fitur Pengambilan Data (Get Data)

Pada fitur ini merupakan fitur yang berfungsi sebagai *eksekutor* (pelaksana) tugas untuk melakukan proses ekstraksi pada halaman *digital library* sesuai dengan jumlah data yang diinginkan. Fitur ini dapat melakukan pencarian skripsi secara *realtime* dengan memasukkan angka nominal pada *textfield* yang telah disediakan, maka sistem akan menampilkan proses pengambilan data yang diekstraksi. Pengguna juga dapat melihat hasil ataupun proses ekstraksi pada *textfield* yang berada dibawah tombol fitur *get data*. Untuk bisa melihat lebih detail lagi terkait hasil ekstraksi pengguna dapat melihatnya pada database yang tersimpan, karena fitur ini memiliki fitur otomatis menyimpan kedalam database ketika proses *get data* berhasil. Tampilan fitur ini bisa dilihat pada **Gambar 4.8** dan **Gambar 4.9** untuk hasil dari *get data*.



Gambar 4.8 Fitur Get Data



Gambar 4.9 Hasil Fitur Get Data

```

public String getJudul() {
    Judul = getData(Data_Html, "<span
class=\"citation_title\">", "</span></td>");
    return Judul;
}

public String getPenulis() {
    Penulis = getData(Data_Html, "<span
class=\"citation_author\">", "</span></td>");
    return Penulis;
}

public String getTahun() {
    Tahun = getData(Data_Html,
"<td><strong>Tahun</strong></td>", "</tr>");
    Tahun = getData(Tahun, "<td><strong>",
"</strong></td>");
    return Tahun;
}

```

Gambar 4.10 Source Code Get Data

4.1.6 Fitur Export Ke RDF (*Resource Description Framework*)

Pada fitur ini merupakan proses terakhir dari semua tahapan aplikasi ini, pada fitur ini berfungsi sebagai *converter* atau sebuah perintah untuk mengubah suatu file menjadi format yang berbeda agar dapat dibaca oleh suatu aplikasi yang dibutuhkan, dalam kasus ini data yang dibutuhkan untuk dikonversi adalah data dari format awal berupa dokumen metadata yang berada pada *digital library* UIN Maliki Malang menjadi format dalam bentuk RDF (*Resource Description Framework*). Proses pengkonversian format ini menggunakan kaidah atau pakem dari RDF yang adopsi dari www.w3.org. Kemudian setelah proses konversi ini berhasil, maka hasil dari format RDF ini akan di uji coba pada tahapan pengujian selanjutnya. Untuk lebih jelasnya fitur ini dapat dilihat pada **Gambar 4.11**



Gambar 4.11 Fitur Export Ke RDF

```

StringBuilder sb = new StringBuilder();
sb.append("<?xml version=\"1.0\"?>\n"
+ "<rdf:RDF\n"
+
"xmlns:rdf=\"http://www.w3.org/1999/02/22-rdf-syntax-ns#\" \n"
+ "xmlns:skripsi=\"http://lib.uin-
malang.ac.id/skripsi#\">");
for (crawData TreeData : FixData) {

    txtStatus.setText("Crawling data from : "
+ TreeData.Link_Data);
    System.out.println(TreeData.toString());

    sb.append("<rdf:Description
rdf:about=\""+TreeData.Link_Data.replace("&", "%26")+"\">\n"
+ "
<skripsi:judul>"+TreeData.Judul+"</skripsi:judul>\n"
+ "
<skripsi:penulis>"+TreeData.Penulis+"</skripsi:penulis>\n"
+ "
<skripsi:tahun>"+TreeData.Tahun+"</skripsi:tahun>\n"
+ "
<skripsi:fakultas>"+TreeData.Fak+"</skripsi:fakultas>\n"
+ "
<skripsi:jurusan>"+TreeData.Jurusan+"</skripsi:jurusan>\n"
+ "
<skripsi:pembimbing>"+TreeData.Pembimbing+"</skripsi:pembimbing>\n"
+ "
<skripsi:keyword>"+TreeData.Keword+"</skripsi:keyword>\n"
+ "</rdf:Description>\n");
}

```

Gambar 4.12 Source Code Fitur Export Ke RDF

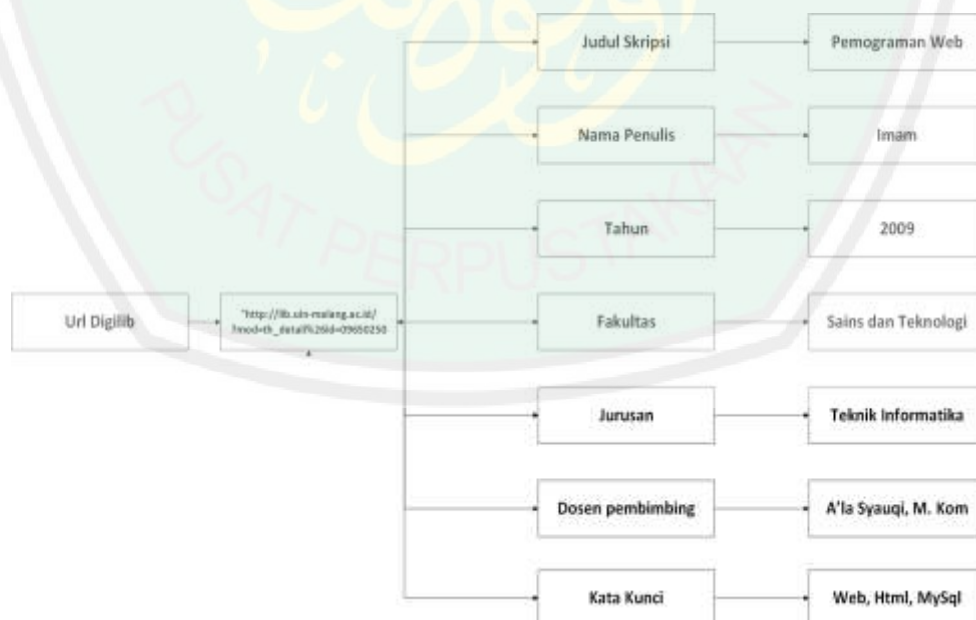
4.2 Implementasi RDF (*Resource Description Framework*)

Pada proses implementasi RDF (*Resource Description Framework*) ini memiliki beberapa tahapan proses untuk bisa menghasilkan sebuah rancangan RDF yang akan dijadikan sumber pengetahuan atau *knowledge base* dari dalam *digital library* UIN Maliki Malang. Tahapan-tahapan tersebut meliputi penentuan *ontology*, pengekstraksian metadata dari halaman *digital library*, penyimpanan hasil ekstraksi ke database kemudian pengkonversian (implementasi) ke format RDF (*Resource Description Framework*) atau lebih familiar dengan format .rdf. Untuk penjelasannya akan dibahas sebagai berikut:

4.3 Perancangan *Ontology*

Pada proses perancangan *ontology* ini dibangun berdasarkan konsep yang telah ditentukan sebelumnya, rancangan *ontology* ini menjadi awal mula dari tujuan perancangan RDF untuk diimplementasikan sebagai *knowledge base digital librar*, sehingga *ontology* harus benar-benar sesuai dengan objek yang akan diteliti karena jika rancangan *ontology* ini tidak sesuai atau sinkron dengan data objek maka hasil akhir (RDF) tidak akan sesuai dengan harapan.

Rancangan *ontology* ini akan dibangun dengan objek pada halaman *digital library* UIN Maliki Malang. Pada halaman *digital library* beberapa data yang akan di jadikan *ontology* seperti Nomor Induk mahasiswa (NIM), Judul Skripsi, Nama Penulis, Tahun, Jurusan, Fakultas, Dosen Pembimbing dan Kata Kunci (*keyword*) dari Abstrak, jadi jika diilustrasikan kedalam sebuah graph akan berupa seperti **Gambar 4.13**

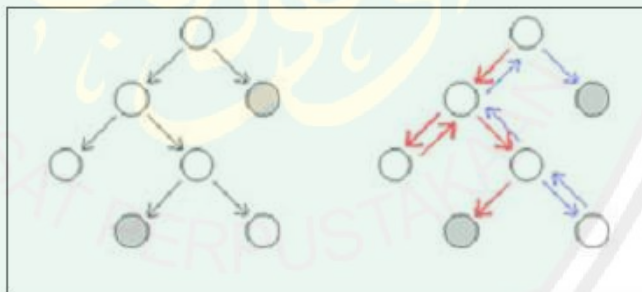


Gambar 4.13 Rancangan *Ontology digital library*

4.3.1 Ekstraksi Metadata Digital Library

Pada proses ekstraksi ini data yang diambil berada pada halaman skripsi *digital library* dengan mengambil dokumen metadata yang diperlukan sesuai dengan perancangan *ontology* sebelumnya. Proses pengeksraksian data dilakukan dengan cara menyeleksi dan menemukan sintaks tertentu yang ada pada halaman html *digital library* UIN Maliki Malang, tag-tag html tersebut dihapus kemudian diambil data sesuai dengan definisi rancangan *ontology* dengan menggunakan kaidah penelusuran dari *algoritma dept first search*.

Algoritma dept first search merupakan sebuah algoritma pencarian dengan menggunakan pencarian dimulai dari root atau akar ke node dengan level n paling kanan, dilanjutkan ke semua anaknya sebelum dilanjutkan pencarian ke node-node selevel. Untuk lebih jelasnya dapat dilihat pada **Gambar 4.14** dibawah ini



Gambar 4.14 Teknik Penelusuran Algoritma Dept First Search

Gambar 4.14 telah menampilkan dengan jelas bagaimana teknik pencarian pada depth-first search. Tanda panah berwarna merah merupakan alur pencarian yang dilakukan pada tahap pertama, kemudian dilanjutkan dengan alur yang ditandai dengan tanda panah berwarna biru.

Secara derkriptif, *algoritma depth-first search* dapat disajikan dalam bentuk sebagai berikut :

Algoritma depth-first search :

1. Jika keadaan awal merupakan tujuan, keluar (sukses).
2. Jika tidak demikian, kerjakan langkah-langkah berikut ini sampai tercapai keadaan sukses atau gagal.
 - a. Bangkitkan succesor E dari keadaan awal. Jika tidak ada succesor, maka akan terjadi kegagalan.
 - b. Panggil depth-first search dengan E sebagai keadaan awal.
 - c. Jika sukses berikan tanda sukses. Namun jika tidak, ulangi langkah-2.

Setelah proses ekstraksi yang dilakukan yaitu dengan mengambil dokumen-dokumen metadata yang ada pada *digital library* secara otomatis data yang telah didapatkan akan langsung tersimpan kedalam database. Untuk hasil dari proses ekstraksi dapat dilihat pada **Gambar 4.15**, dan **Gambar 4.16** untuk potongan source code dari pengambilan data (get data) serta **Gambar 4.17** untuk potongan source code penyimpanan ke database. Sedangkan untuk hasil penyimpanan ke database yang berupa .sql dapat dilihat pada **Gambar 4.18**.



Gambar 4.15 Proses Ekstraksi Dokumen Metadata Digital Library

```
public String getJudul() {
    Judul = getData(Data_Html, "<span
class=\"citation_title\">", "</span></td>");
    return Judul;
}

public String getPenulis() {
    Penulis = getData(Data_Html, "<span
class=\"citation_author\">", "</span></td>");
    return Penulis;
}

public String getTahun() {
    Tahun = getData(Data_Html,
"<td><strong>Tahun</strong></td>", "</tr>");
    Tahun = getData(Tahun, "<td><strong>",
"</strong></td>");
    return Tahun;
}

public String getFak() {
    Fak = getData(Data_Html,
"<td><strong>Fakultas</strong></td>", "</tr>");
    Fak = getData(Fak, "<td><strong>", "</strong></td>");
    return Fak;
}

public String getJurusan() {
    Jurusan = getData(Data_Html,
"<td><strong>Jurusan</strong></td>", "</tr>");
    Jurusan = getData(Jurusan, "<td><strong>",
"</strong></td>");
    return Jurusan;
}
```

Gambar 4.16 Source code get data

```

MyQConn myQConn = new MyQConn(frmSetting);
    for (crawData TreeData : FixData) {

        txtStatus.setText("Crawling data from : " +
TreeData.Link_Data);
        TreeData.Data_Html =
crawlContent(TreeData.Link_Data).toString();
        // System.out.println(TreeData.toString());
        TreeData.doParse();
        if (myQConn.cekQuery("SELECT `ID` FROM
`tb_hasil` WHERE `ID` = \"" + TreeData.ID + "\"")) {

        } else {
            StringBuffer sb = new StringBuffer();
            sb.append("INSERT INTO `tb_hasil`(`ID`,
`ID_jur`,`"
                                + " `Link_Data`, `Judul`,`"
                                + " `Penulis`, `Tahun`,`"
                                + " `Fak`, `Jurusan`,`"
                                + " `Pembimbing`,`"
                                + " `Keyword`,`"
                                + " `Isi`,`"
                                + " `Data_Html`) VALUES "
                                + " (\"" + TreeData.ID + "\"
                                + "\",\"" + TreeData.ID_jur + "\"
                                + "\",\"" + TreeData.Link_Data +
                                + "\",\"" + TreeData.Judul + "\"
                                + "\",\"" + TreeData.Penulis + "\"
                                + "\",\"" + TreeData.Tahun + "\"
                                + "\",\"" + TreeData.Fak + "\"
                                + "\",\"" + TreeData.Jurusan + "\"
                                + "\",\"" + TreeData.Pembimbing +
                                + "\",\"" + TreeData.Keyword + "\"
                                + "\",\"" + TreeData.Isi + "\"
                                + "\",'-' )");
            myQConn.runQuery(sb.toString());
        }
    }

```

Gambar 4.17 Source code penyimpanan ke database

Tabel 4.1 Tabel Database *MySql*


ID	Nama	Deskripsi	Pre-requisite	Other Columns
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
4	...	...	...	...
5	...	...	...	...
6	...	...	...	...
7	...	...	...	...
8	...	...	...	...
9	...	...	...	...
10	...	...	...	...

4.3.2 Rancangan *Resource Description Framework*

RDF adalah suatu model metadata untuk merepresentasikan *resource* yang terdapat di web. Ide dasar RDF adalah mendeskripsikan resource dalam bentuk ekspresi Subyek-Predikat-Obyek (SPO), atau dikenal dengan triple dalam terminology RDF. Subyek adalah entitas yang ditunjukkan oleh teks. Sedangkan predikat adalah komposisi yang menerangkan sudut pandang dari subyek yang dijelaskan oleh obyek. Hal yang menarik dari RDF yaitu obyek dapat menjadi subyek yang nantinya diterangkan oleh obyek yang lainnya. Sehingga obyek atau masukan dapat diterangkan secara jelas dan detail serta sesuai dengan keinginan pengguna yang memberikan masukan.

Fathul Wahid menulis - Dasar-dasar Pemrograman adalah sebuah resource (kalimat) dalam bentuk ekspresi Subyek Predikat Obyek yang masingmasing Subyek adalah entitas. Dan setiap entitas mempunyai identitas yaitu URI (*Uniform Resource Identifier*). Masing-masing resource disebut triple yang

merepresentasikan sebuah informasi atau *knowledge*, sehingga komputer bisa memahami informasi yang ada di web.

RDF (*Resorce Description Framework*) merupakan tujuan atau hasil akhir dari semua proses yang ada pada aplikasi digilib ekstraktor ini, semua tahapan proses mulai dari akan bermuara pada sebuah format yaitu *.rdf*. sebelumnya untuk merepresentasikan semua data yang ada pada *digital library*, RDF akan dibuat berdasarkan Dari skema *ontology* kemudian dibuat suatu bentuk RDF data yang ada dengan menggunakan kaidah dalam kasus ini peneliti menggunakan skema RDF dari www.w3.org. Untuk penyusunannya memakai konsep XML, kemudian XML tersebut ditanamkan ke dalam aplikasi ini sehingga aplikasi akan otomatis menkonversi data menjadi *.rdf*. untuk lebih jelasnya terkait syntax dari *digital library* dapat dilihat pada **Gambar 4.18**.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:skripsi="http://lib.uin-
malang.ac.id/skripsi#"><rdf:Description rdf:about="http://lib.uin-
malang.ac.id/?mod=th_detail%26id=07140059">
  <skripsi:judul>Penerapan Contextual Teaching and Learning (CTL)
Dalam Pembelajaran Matematika Di SDI Surya Buana
Malang.</skripsi:judul>
  <skripsi:penulis>RIJALI WAHDI</skripsi:penulis>
  <skripsi:tahun>2009</skripsi:tahun>
  <skripsi:fakultas>Tarbiyah</skripsi:fakultas>
  <skripsi:jurusan>Pendidikan Guru Madrasah
Ibtidaiyah</skripsi:jurusan>
  <skripsi:pembimbing>Drs. H. Asmaun Sahlan .M.Ag..
</skripsi:pembimbing>
  <skripsi:keyword>Contextual Teaching and Learning (CTL),
Pembelajaran Matematika</skripsi:keyword>
</rdf:Description>
</rdf:RDF>
```

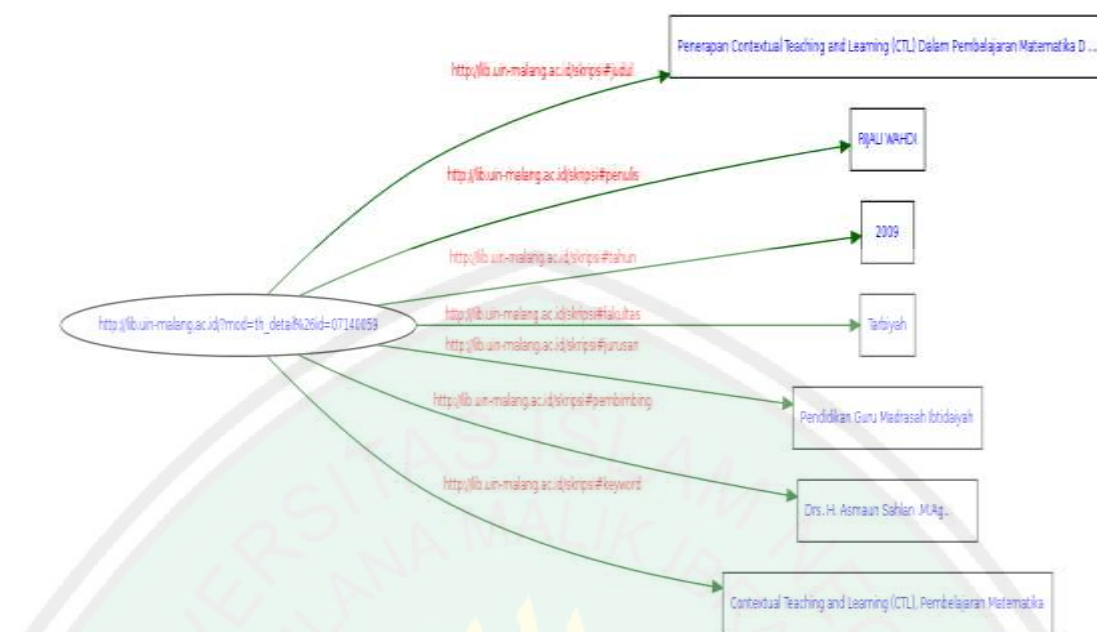
Gambar 4.18 Sintak RDF Digial Library

Format awal pada dokumen di atas adalah suatu tanda bahwa dokumen tersebut berupa dokumen RDF dengan konsep penulisan secara XML yang mengacu pada dokumen yang telah ditunjuk oleh prefix (alamat yang ditunjuk berupa alamat URI, Universal Resource Identifier) yang bertujuan agar tag-tag yang ada dalam dokumen dapat diterjemahkan sesuai dengan konsep yang telah disepakati pada awal dokumen tersebut.

Setelah RDF selesai buat berdasarkan penulisan diatas, maka kemudian format RDF akan dideskripsikan menjadi suatu resource dalam bentuk ekspresi Subyek-Predikat-Obyek (SPO) kemudian akan diinterpretasikan menjadi model sebuah *graph-graph* sesuai dengan ekspresi dari Subyek-Predikat-Obyek (SPO) yang ada pada digital library UIN Maliki Malang. Untuk detilnya dapat dilihat pada **Gambar 4.19** dan **Gambar 4.20**.

Subject	Predicate	Object
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3d1	"Penerapan Contextual Teaching and Learning (CTL) Dalam Pembelajaran Matematika Di SDI Surya Buana Malang."
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3dpemulis	"RIJALI WANDI"
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3dtahun	"2009"
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3dfamiliat	"Tarbiyah"
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3djurusan	"Pendidikan Guru Madrasah Ibtidaiyah"
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3dpewhiring	"Drs. H. Asmaun Sahlan .M.Ag.."
http://lib.uin-malang.ac.id/7wmdeth_detail%3d%3d0714003	http://lib.uin-malang.ac.id/ekripsi%3d%3dkeyword	"Contextual Teaching and Learning (CTL), Pembelajaran Matematika"

Gambar 4.19 SPO dari Digital Library



Gambar 4.20 Skema *graph* dari *Digital Library*

4.4 Uji Coba

Setelah RDF yang digunakan untuk menyimpan informasi dari setiap skripsi terbentuk, maka untuk mengambil informasi dari RDF tersebut diperlukan sebuah query SPARQL.

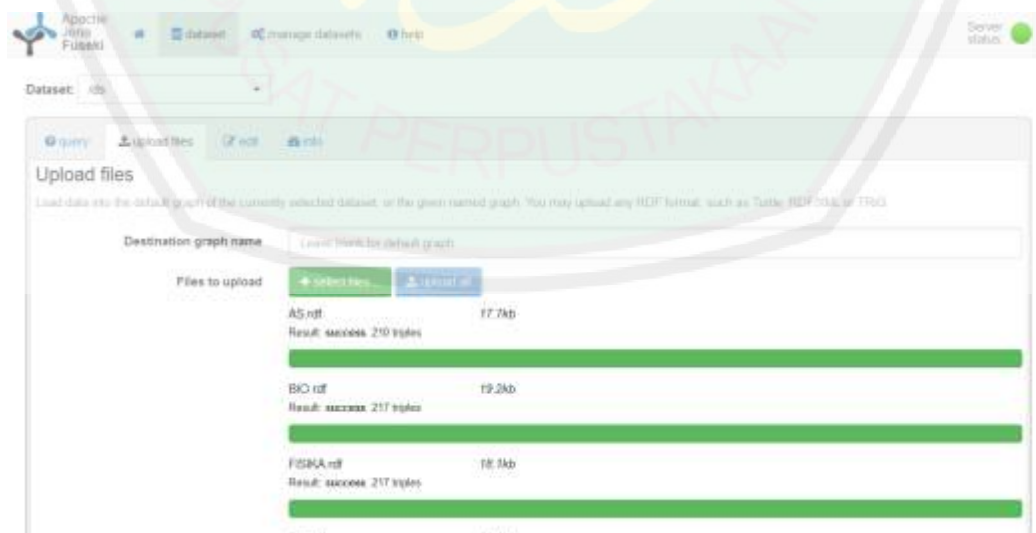
Model data RDF berupa suatu statemen dalam bentuk triple yang terdiri dari subyek, predikat, dan obyek (SPO). Untuk mendapatkan informasi dari suatu *graph* RDF dibutuhkan suatu query. SPARQL merupakan suatu bahasa query yang dapat digunakan untuk mengakses data pada pengolahan data selanjutnya. Dengan menggunakan SPARQL memungkinkan untuk :

- a. Mengambil nilai dari data yang terstruktur maupun data yang semi terstruktur.

- b. Mengembangkan data dengan melakukan query terhadap suatu relasi yang tidak diketahui.
- c. Dapat melakukan query operasi join yang kompleks pada database yang berlainan secara lebih sederhana.
- d. Mengubah suatu data RDF menjadi vocabulary yang lain.

Hasil dari query SPARQL dapat mengembalikan nilai dalam beberapa format data yang antara lain : XML, JSON, RDF, dan HTML.

Untuk pengujian dari hasil perancangan RDF ini, digunakan query SPARQL select untuk menampilkan informasi tersebut dengan menggunakan tool fuseki server sebagai media pengujiannya. Sebelum proses query dilakukan, data yang hasil .rdf diupload dulu ke server fuseki agar data RDFnya bisa dibaca oleh query dari fuseki server. **Gambar 4.21** adalah proses upload file .rdf ke server fuseki server.



Gambar 4.21 Proses upload RDF ke fuseki server

Selanjutnya pada **Gambar 4.22** di bawah ini adalah salah satu query SPARQL yang digunakan untuk menampilkan semua skripsi dari Jurusan Teknik Informatika.

```
PREFIX skripsi: <http://lib.uin-malang.ac.id/skripsi#>
select ?judul ?penulis ?tahun ?jurusan where
{
  ?x skripsi:judul ?judul.
  ?x skripsi:penulis ?penulis.
  ?x skripsi:tahun ?tahun.
  ?x skripsi:jurusan ?jurusan.
  FILTER regex(?jurusan, "Teknik Informatika") .
}
```

Gambar 4.22 Query untuk menampilkan jurusan Teknik Informatika

Kemudian dari proses query diatas akan menampilkan semua data informasi tentang semua skripsi dari jurusan Teknik Informatika. Untuk hasilnya bisa dilihat pada **Gambar 4.23**

	judul	penulis	tahun	jurusan
1	"Rancang Bangun Sistem Informasi Akademik Online Universitas Islam Negeri Malang"	"Ayo Hanani"	"2009"	"Teknik Informatika"
2	"Rancang Bangun Sistem Pembelajaran Jarak Jauh (e-learning) di Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang"	"Habibur Rohman"	"2010"	"Teknik Informatika"
3	"Rancang Bangun Aplikasi Sistem Pakar Mendiagnosa Secara Diri Kanker Kandungan"	"TITI MAMUHAH"	"2009"	"Teknik Informatika"
4	"Rancang Bangun Sistem Informasi Industri Mebel Ukir Jepara Berbasis Web"	"Ahmad Wahyudi"	"2010"	"Teknik Informatika"
5	"SPK untuk mencari laba maksimal pada penjualan tutup LPG"	"M. Taufik Chukomdi"	"2010"	"Teknik Informatika"
6	"Function Structure Plant Model Pertumbuhan Tanaman Bunga Chrysanthemum Indicum Pink Terhadap Pengaruh Pemberian Pupuk MKP Berbasis XL-System"	"Fahrazul Zuhri"	"2011"	"Teknik Informatika"
7	"Sistem Informasi Manajemen Pabrik Menggunakan Webcam sebagai Pendeteksi Kebenaran Model"	"Ali Masyrud"	"2010"	"Teknik Informatika"

Gambar 4.23 Hasil Query untuk menampilkan semua skripsi jurusan Teknik Informatika

Selanjutnya pada **Gambar 4.24** di bawah ini adalah salah satu query SPARQL yang digunakan untuk menampilkan semua skripsi dari Fakultas Psikologi.

```
PREFIX skripsi: <http://lib.uin-malang.ac.id/skripsi#>
select ?judul ?penulis ?tahun ?jurusan where
{
  ?x skripsi:judul ?judul.
  ?x skripsi:penulis ?penulis.
  ?x skripsi:tahun ?tahun.
  ?x skripsi:jurusan ?jurusan.
  FILTER regex(?jurusan, "Psikologi") .
}
```

Gambar 4.24 Query untuk menampilkan skripsi dari Fakultas Psikologi

Selanjutnya hasil informasi yang diperoleh dari proses query pada **Gambar 4.21** dapat dilihat pada **Gambar 4.25**

	judul	penulis	fakultas	jurusan
1	"Hubungan Antara Profesionalitas Guide Dengan Kepuasan Konsumenten"	"Ahmad Jihadi Khoiri"	"Psikologi"	"Psikologi"
2	"Perbedaan Penilaian Terhadap Hukuman Pelaku Pemerkosaan Ditinjau Dari Jenis Kelamin"	"Rifa'atul Mahmudah"	"Psikologi"	"Psikologi"
3	"PENGARUH PENCIPTAAN SUASANA RELIGIUS TERHADAP KESEHATAN MENTAL DI PANTI ASUHAN ZAINUDDIN SEPANJANG SIDOARJO"	"Hidayatul Azizah"	"Psikologi"	"Psikologi"
4	"Penerapan Full Day School dalam Meningkatkan Kreativitas Siswa MTs Surya Buana Malang"	"Baiq Farida Aprianita"	"Psikologi"	"Psikologi"
5	"Hubungan Religiusitas Dengan Perilaku agresif Remaja Madrasah Tsanawiyah Persiapan Negeri Batu"	"Ratna Mulidhah Effendi"	"Psikologi"	"Psikologi"

Gambar 4.25 Query untuk menampilkan semua skripsi dari Fakultas Psikologi

Proses pengujian pada aplikasi ini yaitu dengan melakukan keakuratan hasil *sharing* data pada aplikasi ini menggunakan *query sparql*. *Query sparql* adalah sebuah perintah untuk mengquerikan data dengan menggunakan aplikasi *tool jena fuseki server*, namun untuk dapat mendapatkan data *query* yang diinginkan diperlukan sebuah proses upload hasil perancangan data RDF, data RDF ini berisi data yang akan diuji dengan *tool jena fuseki*.

Hasil data *query* dari *tool jena fuseki* ini kemudian akan diperoleh data berupa data *knowledge base* dari *digital library*. Kemudian untuk mengukur hasil ketepatan akurasi data hasil pembuatan *knowledge base* yang diperoleh dapat menggunakan rumus

$$\text{Hasil Akurasi} = \frac{\sum \text{Data Fuseki}}{\sum \text{Data RDF}} \times 100\% \quad (1)$$

Untuk lebih jelasnya hasil pengujian aplikasi digunakan untuk mengetahui keberhasilan proses dari pembuatan *knowledge base digital library* yang diambil dari *resource description framework* dapat dilihat pada tabel 4.2.

Tabel 4.2 Pengujian Pembuatan Knowledge Base Digital Library

No	Kata Kunci Pencarian	Jumlah Data pada RDF	Hasil Pencarian Apache Jena Fuseki
1	Teknik Informatika	190	190
2	Psikologi	70	70
3	Kimia	81	81
4	Fisika	31	31
5	Matematika	20	20
6	Biologi	31	31
7	Ahwal Syakhshiyah	41	29
8	2010	260	260
9	Sistem	80	80
10	Guru	45	45
11	Siswa	81	81
12	Pendidikan Agama Islam	67	50
13	S2 Pendidikan Agama Islam	17	17
14	Suhu	6	6
15	Ruang	3	3

Kemudian pada tabel 4.1 dihitung menggunakan rumus akurasi sehingga menghasilkan :

$$\begin{aligned}
 \text{Hasil Akurasi} &= \frac{\sum \text{Data Fuseki}}{\sum \text{Data RDF}} \times 100\% \\
 &= \frac{190+70+81+31+20+31+29+260+80+45+81+50+17+6+3}{190+70+81+31+20+31+41+260+80+45+81+67+17+6+3} \times 100\% \\
 &= \frac{994}{1023} \times 100\% \\
 &= 97\%
 \end{aligned}$$

Jadi tingkat keakurasian yang didapatkan sebesar **97 %**.

4.5 Integrasi Al-Qur'an dan Hadits

Setiap manusia dibekali panca indra dan kemampuan berpikir. Hal ini tidak lain mempersiapkan dirinya untuk belajar agar memperoleh ilmu pengetahuan, kemahiran, dan ketrampilan. Semua itu sebagai satu persiapan untuk megarungi kehidupan di dunia hingga ia mampu mencapai kesempurnaan insan yang dikaruniakan Allah Swt. Dalam rangka belajar ini, jelas sangat membutuhkan petunjuk yang datang dari luar. Artinya, karena sebuah ilmu pengetahuan itu memerlukan kejelasan, maka langkah belajar ini sangat memerlukan dukungan dari luar, misalnya seorang guru, orang tua, pengalaman atau bahkan penelitian yang akan menunjukkan satu bukti dan kebenaran dari pengetahuannya.

Penggunaan *knowledge base* adalah salah satu cara untuk mempermudah penyebaran ilmu pengetahuan khususnya dalam bidang *sharing data digital library*.

Al-Qur'an sangat mengagungkan kedudukan ilmu dengan pengagungan yang tidak pernah ditemukan bandingannya dalam kitab-kitab suci yang lain. Sebagai bukti, al-Qur'an memberikan sifat kepada bangsa Arab pada masa pra Islam dengan sebutan jahiliyah (masa kebodohan).

Dan di dalam al-Qur'an terdapat ratusan ayat yang menyebutkan tentang ilmu dan pengetahuan. Di dalam sebagian besar ayat disebutkan kemuliaan dan ketinggian derajat ilmu tersebut.

Untuk mengingatkan manusia terhadap anugerah yang telah diberikan kepadanya, Allah berfirman

يَتَأْتِيهَا الَّذِينَ ءَامَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ وَإِذَا قِيلَ
 أَنْشُرُوا فَأَنْشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ ءَامَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۚ وَاللَّهُ بِمَا تَعْمَلُونَ

خَبِيرٌ

"Hai orang-orang beriman apabila kamu dikatakan kepadamu: "Berlapang-lapanglah dalam majlis", Maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. dan apabila dikatakan: "Berdirilah kamu", Maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. dan Allah Maha mengetahui apa yang kamu kerjakan. (QS Al-Mujaadilah : 11)

Diriwayatkan oleh ibn Abi Hitam dari Muqatil bin Hibban, ia mengatakan bahwa suatu hari yaitu hari Jum'at, Rasulullah Shallallahu 'Alaihi Wasallam berada di Shuffah (tempat belajar para sahabat, serambi masjid Nabawi) mengadakan pertemuan di tempat yang sempit, dengan maksud membicarakan penghormatan terhadap pahlawan Perang Badar.

Beberapa pahlawan Perang Badar itu ternyata terlambat datang, di antaranya Shabit dan Qais, dan terpaksa mereka berdiri di luar Shuffah. Mereka mengucapkan salam, lalu Nabi dan para sahabat pun menjawabnya. Para pahlawan Badar itu pun tetap berdiri, menunggu tempat yang disediakan bagi mereka. Tetapi tidak ada yang memperdulikanya.

Melihat kejadian tersebut Rasulullah Shallallahu 'Alaihi Wasallam menjadi kecewa, lalu menyuruh kepada orang-orang di dekatnya untuk berdiri. Di antara mereka ada yang berdiri, tetapi rasa keenggan nampak di wajah mereka. Apalagi orang-orang munafiq saat itu memberikan reaksi dengan maksud mencela Nabi, sambil memprotes, "Demi Allah, Muhammad tidak adil, ada orang yang

datang lebih dahulu dengan maksud memperoleh tempat duduk, tetapi disuruh berdiri untuk diberikan kepada orang yang datang terlambat”. Lalu turunlah ayat ini....

يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قِيلَ لَكُمْ تَقَسَّعُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ وَإِذَا قِيلَ انشُرُوا فَانْشُرُوا

Artinya : “Wahai orang-orang yang beriman! Apabila dikatakan kepadamu berlapang-lapanglah pada majelis-majelis, maka lapangkanlah, niscaya Allah akan melapangkan bagi kamu. Dan jika dikatakan kepada kamu : Berdirilah !”, maka berdirilah...”

Ayat ini berbicara tentang akhlak di dalam pertemuan (majelis). Asal mulanya memang para sahabat duduk bersama mengelilingi Nabi karena hendak mendengar ajaran-ajaran dan hikmah yang akan Nabi ajarkan. Tentu ada yang datang terlebih dahulu, sehingga tempat duduk bersama itu kelihatan telah sempit. Niscaya karena sempitnya itu, orang yang datang kemudian tidak lagi mendapat tempat. Lalu dianjurkanlah oleh Rasul agar yang telah duduk terlebih dahulu, melapangkan tempat bagi yang datang kemudian.

Sebab pada hakikatnya tempat itu belumlah sesempit apa yang kita sangka. Masih ada tempat lowong, masih ada tempat untuk yang datang kemudian. Sebab itu hendaklah yang telah duduk lebih dahulu, melapangkan tempat bagi mereka yang baru datang itu. Karena yang sempit itu bukan tempat, melainkan hati. Thabi’at mementingkan diri pada manusia sebagai kesan pertama, enggan memberikan tempat kepada yang baru datang itu.

Sama seperti konsep siapa yang disebut orang kaya? Bukan orang yang banyak harta, lalu orang miskin adalah mereka yang sempit hartanya. Padahal Rasulullah Shallallahu ‘Alaihi Wasallam menyebutkan :

لَيْسَ الْغِنَى عَنْ كَثْرَةِ الْعَرَضِ ، وَلَكِنَّ الْغِنَى غِنَى النَّفْسِ

Artinya: “Bukanlah orang kaya diukur dengan banyaknya harta dunia. Namun orang kaya adalah merekayang hatinya selalu merasa cukup.” (H.R. Bukhari dan Muslim).

Syaikh Ibnu Baththol menjelaskan, “Hakikat kekayaan sebenarnya bukanlah dengan banyaknya harta. Karena begitu banyak orang yang diluaskan rezki berupa harta oleh Allah, namun ia tidak pernah merasa puas dengan apa yang diberi. Orang seperti ini selalu berusaha keras untuk menambah dan terus menambah harta. Ia pun tidak peduli dari manakah harta tersebut ia peroleh (apakah halal, syubhat atau halal). Orang semacam inilah yang seakan-akan begitu miskin karena usaha kerasnya untuk terus-menerus memuaskan dirinya dengan harta. Karena itu, hakikat kekayaan yang sebenarnya adalah kaya hati, selalu merasa cukup.

Orang yang kaya hati inilah yang selalu merasa cukup dengan apa yang diberi, selalu merasa *qona’ah* (puas) dengan yang diperoleh dan selalu ridha atas ketentuan Allah. Orang semacam ini tidak begitu tamak untuk menambah harta dan ia tidak seperti orang yang tidak pernah letih untuk terus menambahnya. Kondisi orang semacam inilah yang disebut *ghoni* (yaitu kaya yang sebenarnya).

Ibnu Hajar Al-Atsqolani menerangkan pula, “Orang yang disifati dengan kaya hati adalah orang yang selalu *qona’ah* (merasa puas) dengan rezki yang Allah beri. Ia tidak begitu tamak untuk menambahnya tanpa ada kebutuhan. Ia pun tidak seperti orang yang tidak pernah letih untuk mencarinya. Ia tidak meminta-minta dengan bersumpah untuk menambah hartanya. Bahkan yang terjadi padanya ialah ia selalu ridha dengan pembagian Allah yang Maha Adil padanya. Orang inilah yang seakan-akan kaya selamanya”.

Kemudian, pada bagian akhir ayat ini menjelaskan bahwa Allah akan mengangkat tinggi derajat dan kedudukan orang-orang yang beriman dan orang-orang yang diberi ilmu pengetahuan.

Di sini ayat menjelaskan bahwa orang-orang yang beriman diangkat derajat karena imannya kepada Allah dan Rasul-Nya. Sedangkan orang-orang yang berilmu pengetahuan diangkat derajatnya karena mereka dapat memberikan manfaat (*mashlahat*) kepada orang lain. Ilmu di sini tidak terbatas pada ilmu-ilmu agama saja, seperti aqidah/tauhid, tafsir Al-Quran, fiqh ibadah, dsb. Akan tetapi juga termasuk di dalamnya ilmu-ilmu pengetahuan keduniaan, seperti ilmu geografi, fisika, teknik, kedokteran dll.

Sebab semua ilmu pengetahuan itu pada hakikatnya adalah datang dari Allah Subhanahu Wa Ta'ala, sebagaimana firman-Nya :

اللَّهُمَّ اِنْفَعْنِي بِمَا عَلَّمْتَنِي، وَعَلِّمْنِي مَا يَنْفَعُنِي، وَارْزُقْنِي عِلْمًا تَنْفَعُنِي بِهِ

Artinya : “Ya Allah, berilah aku manfaat dari ilmu yang telah Engkau ajarkan kepadaku, ajarkanlah kepadaku ilmu yang bermanfaat bagiku dan tambahkanlah untukku ilmu yang bermanfaat bagiku.” (H.R. An-Nasa’i).

Oleh karena itu, manakala orang-orang beriman mendapatkan ilmu atau kemudian memperoleh kedudukan atau keuntungan dengan ilmunya. Hal itu bukanlah menjadi tujuannya, dan tidaklah ia pantas bersombong diri atau merasa lebih dengan yang lainnya. Sehingga ia menyepelekan orang lain.

Padahal tidak ada manusia yang sempurna. Ya, mungkin ia hebat dalam ilmu tafsir Al-Quran sekalipun, namun ketika ia memerlukan pakaian, maka ia

memerlukan orang memiliki ilmu membuat pakaian. Demikian pula mungkin ia professor sekalipun di bidang kedokteran. Akan tetapi manakala ia akan membangun rumah, ia tidak bisa membangunnya sendiri dengan keprofesorannya tersebut. Ia pasti memerlukan tukang yang ahli dalam masalah bangunan.

Itulah pentingnya mengedepankan persatuan dan kesatuan, menghargai perbedaan, serta menjunjung tinggi nilai-nilai persaudaraan (*ukhuwwah islamiyyah*), hidup berjama'ah (*bil jama'ah*) dan saling mendamaikan.

Sebagaimana firman-Nya :

إِنَّمَا الْمُؤْمِنُونَ إِخْوَةٌ فَأَصْلَحُوا بَيْنَ أَخَوَيْكُمْ وَاتَّقُوا اللَّهَ لَعَلَّكُمْ تُرْحَمُونَ

Artinya : “Sesungguhnya orang-orang mu'min adalah bersaudara karena itu damaikanlah antara kedua saudaramu dan bertakwalah kepada Allah supaya kamu mendapat rahmat”. (Q.S. Al-Hujurat [49] : 10).

Karena itu, tidak lain ucapan orang-orang beriman dengan ilmunya itu adalah memuji Allah :

سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ الْحَكِيمُ

Artinya : “Maha Suci Engkau, tidak ada yang kami ketahui selain dari apa yang telah Engkau ajarkan kepada kami; sesungguhnya Engkaulah Yang Maha Mengetahui lagi Maha Bijaksana”. (Q.S. Al-Baqarah [2] : 32).

Sehingga dengan ilmunya itu, ia akan berusaha mengamalkannya dengan cara menshadaqahkannya kepada orang lain yang memerlukan. Sehingga bermanfaatlah ilmu itu, dan menjadi tambahan pahala buatnya.

Di dalam hadits disebutkan bahwa ilmu yang manfaat merupakan harta warisan yang sangat berharga dan akan terus mengalirkan pahala kepada yang punya, walaupun ia telah meninggal dunia.

إِذَا مَاتَ الْإِنْسَانُ انْقَطَعَ عَنْهُ عَمَلُهُ إِلَّا مِنْ ثَلَاثَةٍ إِلَّا مِنْ صَدَقَةٍ جَارِيَةٍ أَوْ عِلْمٍ يُنْتَفَعُ بِهِ أَوْ وَلَدٍ صَالِحٍ يَدْعُو لَهُ

Artinya : “Jika seorang manusia meninggal dunia, maka terputuslah segala amalannya, kecuali tiga perkara, yaitu : (1) Shadaqah jariyah, dan (2) ilmu yang bermanfaat, dan (3) anak yang shalih yang mendoakannya.” (H.R. Muslim).

Berkaitan dengan ayat di atas menjadi landasan, bahwa ilmu pengetahuan adalah seluruh usaha sadar untuk menyelidiki, menemukan, dan meningkatkan pemahaman manusia dari berbagai segi kenyataan dalam alam manusia. Segi-segi ini dibatasi agar dihasilkan rumusan-rumusan yang pasti. Ilmu memberikan kepastian dengan membatasi lingkup pandangannya, dan kepastian ilmu-ilmu diperoleh dari keterbatasannya. Kata ilmu dalam bahasa Arab "ilm" yang berarti memahami, mengerti, atau mengetahui. Dalam kaitan penyerapan katanya, ilmu pengetahuan dapat berarti memahami suatu pengetahuan, dan ilmu sosial dapat berarti mengetahui masalah-masalah sosial, dan sebagainya. Ruang lingkup ilmu pengetahuan saat ini berkembang luas meliputi berbagai bidang keilmuan, antara lain, teknologi komputasi, sistem informasi, ilmu sosial terapan, maupun ilmu alam. Masing-masing ilmu tersebut seperti berlomba dalam memberikan manfaat kepada manusia dalam mempermudah penyelesaian masalahnya. Informasi dalam dunia ilmu pengetahuan adalah salah satu kunci dan sebagai petunjuk untuk memperoleh suatu pengetahuan baru.

Di dalam *knowledge base* terdapat sebuah informasi diperoleh dari sebuah *resource description framework* yang di dalamnya terdapat sebuah pengetahuan, dan tentunya berbagai macam informasi. Ada suatu pepatah yang berbunyi: “*Siapa menguasai informasi dia menguasai dunia*”. Pepatah tersebut memiliki makna yang mendalam. Dunia informasi dan kebutuhan akan berita merupakan suatu keniscayaan yang tidak terbantahkan. Siapapun yang mengetahui informasi terlebih dahulu akan dapat memperoleh hasil yang lebih baik daripada orang lain. Yang menguasai informasi, menguasai ilmu pengetahuan. Barang siapa yang mempunyai ilmu pengetahuan, Allah meninggikan derajat dibandingkan dengan orang yang tidak berilmu pengetahuan.

BAB V

PENUTUP

5.1 Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa peneliti berhasil membuat sebuah aplikasi ekstraksi data digital library untuk perancangan RDF guna pembuatan *knowledge base digital library*. Pembuatan *knowledge base digital library* dapat dilakukan dengan menggunakan hasil dari perancangan *resource description framework*. Untuk mengakses metadata pada file Perancangan *resource description framework* dapat dikonversikan dalam bentuk N-Triple table yang berpola subjek-predikat-objek.

akan tetapi berdasarkan *pengujian knowledge base* belum seluruh mendapat hasil yang maksimal. Hal ini dikarenakan adanya kesamaan data RDF pada jurusan S2 Pendidikan Agama Islam dengan Pendidikan Agama Islam sehingga hasil akurasi yang diperoleh adalah 97 %.

5.2 Saran

Pada aplikasi ini memiliki kelemahan yaitu memerlukan koneksi internet yang cepat, karena aplikasi ini memerlukan pengambilan data yang banyak sehingga dibutuhkan koneksi yang cepat agar proses pengambilan data dari *Digital Library* UIN Maliki Malang berjalan dengan baik.

DAFTAR PUSTAKA

- Andi Nugroho Moh. 2013. *Ensiklopedia Objek Wisata Jawa Timur Menggunakan Semantic Web*.
- Bernard Renaldy dkk. 2008. *Ontology e-Learning Content Berbasis Web Semantic*. Yogyakarta: Universitas Gadjah Mada.
- Berners-Lee, T., Hendler, J., Lasilla, O.. 2001. *The Semantic Web*. American Scientific.
- Bing Liu, and Yanhong Zhai. 2007. *Extracting Web Data Using Instance-Based Learning*. University of Illinois at Chicago.
- Bing Liu. 2005. *Web Content Mining*. University of Illinois at Chicago. <http://www.frenchlane.com/Web-ContentMining-4.pdf>,
- Dian Farisah Ristanti. 2013. *Mesin Pencari Berbasis Semantic Search Menggunakan Algoritma Boyer-Moore pada Ensiklopedia Masjid Bersejarah Di Indonesia* Bing Liu, Robert Grossman, and Yanhong Zhai, Mining Data Records in Web Pages, University of Illinois at Chicago, 2003, http://www.cs.uic.edu/~liub/publications/kdd_2003-dataRecord.pdf,
- Fitrya Nadya Sabyla. 2012. *Perancangan Dan Implementasi Search Engine Menggunakan Teknologi Semantic Web Pada Aplikasi Digital Library*.
- Kamus Besar Bahasa Indonesia*. 1996. Jakarta: Depdikbud.
- Nadya Sabyla Fitrya. 2012. *Perancangan Dan Implementasi Search Engine Menggunakan Teknologi Semantic Web Pada Aplikasi Digital Library*. Tersedia di http://lib.uin-malang.ac.id/?mod=th_detail&id=07650026 diakses pada 27 februari 2014.
- Nava'atul Fadillah, Novrido Charibaldi, Herlina Jayadianti. 2010. *Penerapan Teknologi Semantic Web Pada Aplikasi Pencarian Koleksi Perpustakaan (Studi Kasus: Perpustakaan Fti Upn "Veteran" Yogyakarta)*. Tersedia di http://repository.upnyk.ac.id/461/1/DAFTAR_ISI.pdf diakses pada 27 februari 2014.
- Nisa Hidayani, Juni Nurma Sari, Rahmat Suhatman. 2013. *Perancangan dan Implementasi Algoritma Brute Force Untuk Pencarian String Pada Website PCR*. Tersedia di https://journal.pcr.ac.id/student-journal/page/read_pdf.php?name=Jurnal_NisaHidayani.pdf&id=57 diakses pada 27 februari 2014.
- Pollock, Jefray T. 2009. *Semantic Web for Dummies*. Indiana Polis: Wiley Publishing Suteja.

- Riyanarto Sarno dan Faisal Rahutomo. 2009. *Penerapan Algoritma Weighted Tree Similarity Untuk Pencarian Semantik*. Tersedia di http://digilib.its.ac.id/public/ITS-Master-11561-5107201001-Approval_Sheet.pdf diakses pada 1 Oktober 2013.
- Saryo, Riyanarto, dkk. 2012. *Semantic Search: Pencarian berdasarkan konten*. Yogyakarta:Penerbit ANDI.
- Sugeng Listyo Prabowo. 2014. *Cita-cita Besar Kami adalah Menuju World Class University*. Diakses dari <http://sugeng.lecturer.uin-malang.ac.id/2014/04/cita-cita-besar-kami-adalah-menuju-world-class-university-3/>; Diakses pada 10 Mei 2014.
- Wahyono, Teguh. 2004. *Sistem Informasi (Konsep Dasar, Analisis Desain dan Implementasi)*. Yogyakarta: Graha Ilmu.
- Breadth First Search/Depth First Search Animations.
http://www.cs.sunysb.edu/~skiena/combinatorica_animations/search.html.
Tanggal Akses : 27 Februari 2015
- Breadth-first search. http://en.wikipedia.org/wiki/Breadth-first_search. Tanggal Akses : 27 Februari 2015
- Cycles. <http://www.geom.uiuc.edu/docs/research/webviz/webviz/node4.html> .
Tanggal Akses : 1 Januari 2008
- Depth-First Search. http://en.wikipedia.org/wiki/Depth-first_search. Tanggal Akses : 27 Februari 2015
- Depth first vs Breadth first Search.
http://www.macs.hw.ac.uk/~alison/ai3notes/paragraph2_6_2_1_0_1.html
. Tanggal Akses : 27 Februari 2015
- Depth First and Breadth First Search by kirupa.
http://www.kirupa.com/developer/actionsript/depth_breadth_search.htm
.Tanggal Akses : 27 Februari 2015
- Goodrich, Tamassia. Breadth-First Search
<http://ww0.java4.datastructures.net/handouts/BFS.pdf> Tanggal Akses : 27 Februari 2015

BIODATA PENELITIAN

Nama Lengkap : Musyfiq Luthfi
 Tempat, Tanggal Lahir : Sumenep, 02 Agustus 1990
 Jenis Kelamin : Laki-laki
 Alamat Asal : Karang Cempaka, Kec. Bluto, Kab. Sumenep
 Alamat Sekarang : Jl. Sumber Sari Gang I Nomer 46, Lowokwaru,
 Kota Malang
 No. Handphone : 081803483377
 Email : mluthfiindonesia@gmail.com
 Facebook : Musyfiq Luthfi
 Twitter : @m_ngalaz

Pendidikan Formal

1995 – 1997 : TK. Nurul Islam Karang Cempaka
 1997 – 2003 : MI Tarbiyatul Athfal
 2003 – 2006 : MTs Nurul Islam
 2006 – 2009 : MA Nurul Islam
 2009 – 2015 : Jurusan Teknik Informatika Fakultas Sains dan Teknologi
 Universitas Islam Negeri Maulana Malik Ibrahim Malang

Pendidikan Non Formal

2009 – 2010 : Ma'had Sunan Ampel Al-Ali Universitas Islam Negeri
 Maulana Malik Ibrahim Malang
 2009 – 2010 : Program Perkuliahan Khusus Bahasa Arab (PKPBA)
 Universitas Islam Negeri Maulana Malik Ibrahim Malang
 2010 – 2011 : Program Perkuliahan Khusus Bahasa Inggris (PKPBI)
 Universitas Islam Negeri Maulana Malik Ibrahim Malang

Pengalaman Organisasi

- Anggota Pergerakan Mahasiswa Islam Indonesia (PMII) Rayon Galileo
 Fakultas Sains dan Teknologi tahun 2009

- Departemen Gerakan Pergerakan Mahasiswa Islam Indonesia (PMII) Rayon Galileo Fakultas Sains dan Teknologi tahun 2010-2011
- DirJend keorganisasian Badan Eksekutif Mahasiswa (BEM) Fakultas Sains dan Teknologi UIN Maliki Malang tahun 2010
- Departemen Pengkaderan Pergerakan Mahasiswa Islam Indonesia (PMII) Rayon Galileo Fakultas Sains dan Teknologi tahun 2011-2012
- Sekretaris I Himpunan Mahasiswa Jurusan Teknik Informatika UIN Maliki Malang tahun 2011
- Menteri Dalam Negeri Dewan Eksekutif Mahasiswa (DEMA) Fakultas Sains dan Teknologi UIN Maliki Malang tahun 2012
- Departemen Gerakan Pergerakan Mahasiswa Islam Indonesia (PMII) Komisariat Sunan Ampel UIN Maliki Malang tahun 2012-2013
- Kementrian Seni dan Budaya Dewan Eksekutif Mahasiswa (DEMA) UIN Maliki Malang tahun 2013-2014

Aktivitas dan Pelatihan

- Sertifikat Bahasa Arab
- Sertifikat Bahasa Inggris
- Sertifikat sebagai peserta Capacity Building Pengurus HMJ
- Sertifikat sebagai peserta Capacity Building Pengurus DEMA Fakultas

Demikian Daftar Riwayat Hidup ini dibuat sebagai bahan pertimbangan.

Malang, 30 Juni 2015

Musyfiq Luthfi