

**PERANGKAT LUNAK MESIN PENCARIAN MENGGUNAKAN
METODE *CONCEPTUAL FUZZY SETS***

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

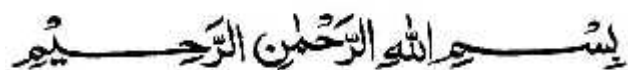
oleh :

AHMAD HAFIZ
10451025494



FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011

KATA PENGANTAR



Assalammu'alaikum wr wb.

Alhamdulillah rabbil'alamin, , puji dan syukur penulis ucapkan kepada Allah SWT atas segala rahmat dan hidayah yang diberikan-Nya, sehingga penulis dapat melaksanakan dan menyelesaikan tugas akhir ini dengan baik. Shalawat beriring salam diucapkan untuk junjungan kita Rasulullah Muhammad SAW, karena jasa Beliau yang telah membawa manusia dari zaman kebodohan ke zaman yang penuh ilmu pengetahuan seperti sekarang ini.

Tugas akhir yang berjudul **PERANGKAT LUNAK MESIN PENCARIAN MENGGUNAKAN METODE *CONCEPTUAL FUZZY SETS*** ini disusun sebagai satu syarat untuk mendapatkan gelar kesarjanaan pada jurusan Teknik Informatika Universitas Islam Negeri Sultan Syarif Kasim Riau.

Dalam penulisan dan penyusunan laporan tugas akhir ini penulis tidak terlepas dari bantuan berbagai pihak yang telah memberikan masukan berupa kritik, saran, motivasi dan dorongan yang sangat bermanfaat bagi penulis. Untuk itu dalam kesempatan ini penulis menyampaikan terima kasih yang tak terhingga kepada :

1. Bapak Prof. DR. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Hj. Yenita Morena, M.Si, selaku Ketua Sidang dan Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Bapak Novriyanto, ST, M.Sc, selaku Ketua Jurusan Teknik Informatika.
4. Bapak Jasril, S.Si, M.Sc, selaku Pembimbing Akademis Penulis.
5. Bapak Muhammad Fikry, S.T, M.Sc selaku Pembimbing I tugas akhir yang selalu sabar dan meluangkan waktu untuk membimbing, memberikan saran dan kritik dalam penyusunan tugas akhir ini.

6. Ibu Elin Haerani, S.T selaku Pembimbing II tugas akhir yang selalu bersedia menerima curahan kesudahan hati penulis selama masa pelaksanaan dalam menyusun tugas akhir ini.
7. Ibu Fitri Wulandari, S.Si, M.Kom, selaku Penguji I tugas akhir yang memberi kritik membangun bagi penulis untuk menyempurnakan pelaksanaan penulisan dalam tugas akhir ini.
8. Bapak Alwis Nazir, S.Kom. M.Kom, selaku Penguji II tugas akhir yang telah memberikan kritik membangun dan saran-saran dalam penyempurnaan tugas akhir ini.
9. Ibu Rice Novita, S.Kom, M.Kom, selaku Koordinator Tugas akhir Jurusan Teknik Informatika.
10. Seluruh dosen dan staf Fakultas Sains dan Teknologi khususnya pada Jurusan Teknik Informatika. Terima kasih atas ilmu yang telah diberikan
11. Papa dan Mama yang tercinta, yang merawat dan membesarkan penulis, untuk usaha dan kerja keras dalam mewujudkan cita-cita penulis, untuk setiap tetes keringat yang diberikannya guna keberhasilan penulis, serta untuk setiap do'a dan air mata yang mengalir, penulis mintakan syurga atas segala pengorbanannya.
12. Paman dan bibiku Prof. DR. H. Alaidin Koto, MA dan Dra. Yustinar Yasin, terima kasih atas perhatian dan usaha serta jasanya, semoga Allah memberikan tempat terindah di-Syurga.
13. Abangku yang tercinta Surya Elhadi, ST adalah abang dan calon pemimpin yang terbaik bagiku, kakakku Laily Rahmawati, S.Pi yang sangat aku cintai, yang tidak hentinya menjaga dan memperhatikanku, dan adikku yang ku sayangi Azizia Oktaviana, engkau adikku satu-satunya yang paling aku banggakan
14. Nenekku Hj. Nurbaya dan Aminah yang senantiasa memberikan doa dan ridhonya.
15. Bibiku Desi Riawati, S.Sos, dan adik-adikku Narul Izza Mutiara Aldin, Narul Auliya Mutiara Aldin, Nayla Rezkia, Dang Merdu Berlian Aldin dan Anugerah Mauliana.

16. Abang Jusnasri, S.T selaku senior Teknik Informatika yang senantiasa bersedia meluangkan waktu bagi penulis dalam menimba ilmu dan belajar menembangkan perangkat lunak berbasis web untuk tugas akhir ini.
17. Sahabatku Fadhilah Syafria, S.T terima kasih atas perhatian, bantuan dan dukungan yang selalu diberikan kepada penulis.
18. Teman-teman seperjuangan angkatan 2004 di Jurusan Teknik Informatika yang telah banyak memberikan bantuan tenaga dan pikirannya untuk kesempurnaan penyusunan makalah ini.
19. Seluruh pihak yang belum penulis cantumkan, terima kasih atas dukungannya.

Penulis menyadari bahwa dalam penulisan laporan ini masih banyak kesalahan dan kekurangan, oleh karena itu kritik dan saran yang sifatnya membangun penulis harapkan untuk kesempurnaan laporan ini. Akhirnya penulis berharap semoga laporan ini dapat memberikan sesuatu yang bermanfaat bagi siapa saja yang membacanya. Amin.

Wassalamu 'alaikum wr.wb.

Pekanbaru, 28 Januari 2011

AHMAD AFIZ

PERANGKAT LUNAK MESIN PENCARIAN MENGUNAKAN METODE *CONCEPTUAL FUZZY SETS*

AHMAD HAFIZ
10451025494

Tanggal Sidang : 28 Januari 2011
Tanggal Wisuda : 24 Februari 2011

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Conceptual Fuzzy Sets merupakan pengembangan dari *Fuzzy Sets Theory*. *Conceptual fuzzy sets* bekerja dengan cara mengekspansi kata - kata kunci frasa utama menjadi beberapa kata kunci frasa lain dengan memberi nilai untuk tiap kata kunci frasa. Kata kunci frasa yang memiliki nilai paling tinggi berarti memiliki nilai relevansi yang paling kuat dengan kata kunci frasa utama.

Perangkat lunak mesin pencarian ini menggunakan metode *conceptual fuzzy sets* untuk memberikan informasi kata kunci tambahan kepada mesin pencari dalam menentukan kata kunci-kata kunci yang memiliki kesamaan makna dengan kata kunci utama yang dimasukkan oleh publik dalam mencari informasi. Kata kunci yang telah diperoleh akan dilakukan perhitungan nilai relevansinya antara sesama kata kunci menggunakan *conditional probability relation* dan terhadap halaman web untuk memperoleh bobot nilai halaman web yang mengandung kata kunci menggunakan Tf-Idf (*Term Frequency - inverse document frequency*). Proses pencarian dalam Perangkat lunak ini akan menghasilkan pencarian yang memiliki nilai atau ranking pada setiap hasilnya. Hasil pencarian ini dapat memberikan informasi yang lebih baik dari mesin pencari lainnya kepada publik dengan memberikan hasil yang terurut dari ranking terbesar, yaitu halaman web yang memiliki kedekatan makna lebih tinggi terhadap kata kunci hingga ranking terendah.

Kata Kunci : *Conceptual Fuzzy Sets*, Kata Kunci, Mesin Pencari, Rangkaian.

SOFTWARE SEARCH ENGINE USING CONCEPTUAL FUZZY SETS

AHMAD HAFIZ
10451025494

Date of Final Exam : January 28th 2011
Date of Graduation : February 24th 2011

Informatics Engineering Departement
Faculty of Sciences and Technology
State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

Conceptual Fuzzy Sets is an extension of Fuzzy Sets Theory. Conceptual fuzzy sets work by expanding the word major keyword phrases into a number of other keyword phrases by giving value for each keyword phrase. keyword phrases that have the highest value means the value relevance of the strongest with the main keyword phrase.

Software search engine based on these keywords using conceptual fuzzy sets to provide information on additional keywords to the search engines in determining the key-word keywords that have the same meaning with the main keywords entered by the user in finding information. Keywords that have been obtained will be calculated relevance value between fellow keywords using conditional probability relations and on the web page to obtain the weight value of web pages containing keywords using Tf-IDF (Term frequency - inverse document frequency). Search process in this software will generate a search that has a value or ranking of each outcome. The results of this search can provide better information than other search engines to the public with memeberikan results are sorted from terebesar ranking web pages that have a higher meaning proximity to keywords to rank the lowest.

Key words : *Conceptual Fuzzy Sets, Keyword, Ranking, Search engine.*

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN.....	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xviii
DAFTAR LAMPIRAN.....	xx
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang.....	I-1
1.2 Rumusan Masalah.....	I-3
1.3 Tujuan Tugas Akhir	I-4
1.4 Batasan Masalah	I-4
1.5 Sistematika Penulisan	I-4
BAB II LANDASAN TEORI.....	II-1
2.1 Web Server	II-1
2.2 Mesin Pencari (Saerch Engine).....	II-4
2.2.1 Cara Kerja Mesin Pencari.....	II-7
2.3 <i>Web Crawler</i>	II-8
2.3.1 Cara Kerja <i>Web Crawler</i>	II-8

2.4	<i>Fuzzy</i>	II-9
2.4.1	Variabel Linguistik	II-10
2.4.2	Fungsi Keanggotaan	II-11
2.4.3	<i>Conceptual Fuzzy Sets</i>	II-12
2.5	<i>Term Frequency – Invers Document Frequency (TF-IDF)</i>	II-15
2.6	<i>Reguler Expression</i>	II-17
BAB III	METODOLOGI PENELITIAN	III-1
3.1	Metodologi Penelitian	III-1
3.1.1	Studi Pustaka	III-2
3.2.2	Identifikasi Masalah	III-2
3.2.3	Analisa Sistem	III-3
3.2.4	Perancangan Sistem	III-3
3.2.5	Implementasi	III-3
3.2.6	Pengujian	III-4
BAB IV	ANALISA DAN PERANCANGAN	IV-1
4.1	Analisa Sistem	IV-1
4.1.1	Analisa Sistem Lama	IV-1
4.1.2	Analisa Sistem Baru	IV-2
4.1.2.1	Kebutuhan Data	IV-3
4.1.2.1.1	Analisa Data <i>Inputan</i>	IV-3
4.1.2.1.2	Analisa Data Keluaran	IV-4
4.1.2.1.1	Analisa Fungsi Sistem	IV-4
4.1.2.2	Analisis <i>Conceptual Fuzzy Sets</i> Dalam <i>Search Engine</i>	IV-5
4.1.2.2.1	<i>Input</i> Kata Kunci	IV-5
4.1.2.2.2	Proses Perhitungan Relevansi	IV-7

4.1.2.2.3	Proses Perhitungan Halaman <i>Web</i> yang Memiliki Kata Kunci dan <i>Extended Keyword</i>	IV-9
4.1.2.2.4	Proses Pengurutan Halaman <i>Web</i>	IV-13
4.1.2.3	Deskripsi Fungsional	IV-14
4.1.2.3.1	<i>Diagram Contexts</i>	IV-14
4.1.2.3.2	<i>Diagram Flow Diagram</i> (DFD).....	IV-15
4.1.2.3.3	Kamus Data.....	IV-19
4.1.2.3.4	Bagan Alir (<i>Flowchart</i>)	IV-20
4.2	Perancangan	IV-21
4.2.1	Perancangan Tabel	IV-21
4.2.1.1	Tabel Admin.....	IV-21
4.2.1.2	Tabel Master <i>URL</i>	IV-21
4.2.1.3	Tabel <i>tbl_URL</i>	IV-21
4.2.2	Perancangan Antar Muka	IV-22
4.2.2.1	Perancangan Antar Muka Untuk Administrator.....	IV-23
4.2.2.1.1	Rancangan <i>Login</i>	IV-23
BAB V	IMPLEMENTASI DAN PENGUJIAN.....	V-1
5.1	Implementasi	V-1
5.1.1	Lingkungan Implementasi.....	V-1
5.1.1.1	Perangkat Keras.....	V-2
5.1.1.2	Perangkat Lunak.....	V-2
5.1.2	Hasil Implementasi.....	V-2
5.2	Pengujian Sistem	V-3
5.2.1	Identifikasi dan Rencana Pengujian	V-3
BAB VI	PENUTUP	VI-1
6.1	Kesimpulan.....	VI-1

6.2 Saran.....	VI-1
----------------	------

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR TABEL

Tabel	Halaman
2.1 HTTP <i>Loader</i>	II-3
2.2 <i>Respons Code Respons Text</i>	II-3
2.3 Gambaran Sederhana <i>Database</i>	II-14
2.4 Pola Umum <i>Regex</i>	II-18
4.1 Jumlah Halaman <i>Web</i>	IV-7
4.2 Jumlah Kemunculan Kalimat Gayus Tambunan Terhadap Halaman <i>Web</i>	IV-9
4.3 Hasil Perhitungan Halaman <i>Web</i> Terhadap Kalimat Gayus Tambunan .	IV-10
4.4 Jumlah Kemunculan Kata Gayus Terhadap Halaman <i>Web</i>	IV-10
4.5 Hasil Perhitungan Halaman <i>Web</i> Terhadap Kata Gayus	IV-11
4.6 Jumlah Kemunculan Kata Tambunan Terhadap Halaman <i>Web</i>	IV-11
4.7 Hasil Perhitungan Halaman <i>Web</i> Terhadap Kata Tambunan	IV-12
4.8 Perhitungan Akhir Untuk Menghasilkan Nilai Relevansi Kata Kunci Terhadap Halaman <i>Web</i> Secara Keseluruhan.....	IV-12
4.9 Pengurutan hasil pencarian halaman <i>web</i> untuk “Gayus Tambunan”	IV-13
4.10 Proses DFD Level 1 Perangkat Lunak Mesin Pencarian Menggunakan Metode <i>Conceptual Fuzzy Sets</i>	IV-16
4.11 Aliran Data DFD Perangkat Lunak Mesin Pencarian Menggunakan Metode <i>Conceptual Fuzzy Sets</i>	IV-16
4.12 DFD Level 2 Proses Pengelolaan <i>Web Administrator</i>	IV-17
4.13 Aliran Data DFD 2 Proses Pengelolaan <i>Web Administrator</i>	IV-17
4.14 Proses DFD Level 2 Proses Master <i>URL</i>	IV-18
4.15 Aliran Data DFD 2 Proses Master <i>URL</i>	IV-18
4.16 Proses DFD Level 2 Proses Pengguna.....	IV-18
4.17 Aliran Data DFD 2 Proses Pengguna.....	IV-18
4.18 Kamus Data admin.....	IV-19
4.19 Kamus Data Master <i>URL</i>	IV-19
4.20 Kamus Data <i>Crawler</i>	IV-19
4.21 Tabel Admin	IV-21

4.22	Tabel <i>master_url</i>	IV-21
4.23	Tabel <i>tbl_url</i>	IV-21
4.24	Deskripsi Perancangan Antarmuka <i>Home</i>	IV-23
5.1	Identifikasi dan Rencana Pengujian.....	V-3

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Komputer memiliki fungsi utama sebagai alat pengolah data menjadi informasi. Komputer merupakan alat bantu kerja yang dapat dikembangkan untuk berbagai hal, salah satu contohnya yaitu pengembangan komputer sebagai alat pencari informasi.

Informasi berita membuat dunia internet semakin berkembang. Perkembangan informasi ini menghadirkan pertumbuhan situs *web* diseluruh dunia meningkat. Dengan pertumbuhan situs-situs *web* di seluruh dunia menghadirkan peranan penting bagi mesin pencari (*search engine*) untuk membantu pengguna internet dalam proses pencarian informasi. Hal ini dikarenakan bagi pengguna yang ingin mendapatkan informasi yang sesuai dengan kebutuhan, maka pengguna itu perlu mengingat setiap *URL* dari *web* yang dibutuhkannya.

Bagian terpenting dari mesin pencari adalah *web crawler*. *Web crawler* adalah *software* yang bertugas mengunjungi situs *web*, membaca isi dari situs *web*, menyimpan informasi ke dalam basis data, dan mengikuti setiap *link-link* yang terdapat pada setiap halaman *web* tersebut. *Web crawler* akan mengunjungi situs *web* secara periodik untuk melihat dan meng-*update* setiap perubahan yang terdapat pada situs *web*.

Sebagian besar mesin pencari yang berkembang pada saat ini seperti Google, dan Yahoo!, dalam melakukan pencarian dan menampilkan informasi

berdasarkan kesesuaian kata kunci dengan menggunakan Metode *Pagerank*. Mesin pencari tersebut mencari halaman *web* yang mengandung kata kunci di dalamnya, dan menampilkan alamat-alamat *URL* dalam hasil pencarian secara terurut sesuai ranking yang telah diberikan oleh mesin pencari. Dapat dikatakan bahwa antara informasi yang didapat dengan kata kunci yang di-*input*-kan memiliki relasi secara sintaks, sehingga hasil pencarian memberikan halaman *web* yang memiliki ranking tertinggi dan mengandung kata kunci yang di-*input*-kan pengguna. Misalkan pengguna meng-*input*-kan kata kunci “Sistem Inventori”, maka yang akan ditampilkan adalah daftar *URL* halaman *web* yang setiap halaman *web*-nya memiliki kata “sistem” atau “inventori” di dalamnya dan terurut berdasarkan ranking tertinggi sampai terendah tanpa memperdulikan tema atau judul dari informasi yang terdapat dari sumber informasi tersebut.

Seluruh mesin pencari memiliki metode masing-masing dalam mencari dan menampilkan hasil informasinya. Salah satunya adalah metode *conceptual fuzzy sets*. *Conceptual fuzzy sets* merupakan suatu metode perhitungan kata dengan pendekatan secara konsep dan makna. Dengan metode ini, suatu pencarian terhadap suatu kata yang di-*input*-kan akan menghasilkan suatu nilai relevansi atau nilai kedekatan makna antara halaman *web* dengan kata kunci. Berdasarkan nilai relevansi yang didapatkan dengan metode perhitungan ini, mesin pencari akan mengurutkan sesuai dengan nilai yang tertinggi sampai nilai yang terendah.

Pengurutan itu dimaksud untuk memberikan alamat halaman *web* yang mengandung kata kunci untuk dapat dikunjungi lebih utama oleh pengguna.

Dalam penelitian tugas akhir ini diharapkan dapat menghasilkan suatu perangkat lunak yang dapat mencari berita teraktual sebagai informasi yang dibutuhkan dan dapat menghasilkan informasi yang lebih baik dari mesin pencari umumnya, karena pencarian halaman *web* didasarkan pada kesamaan makna dengan kata kunci bukan berdasarkan tingginya tingkat ranking suatu halaman *web*. Dengan ini diharapkan pengguna akan memperoleh informasi yang lebih tepat. Mesin pencari ini dibuat dengan Metode *Conceptual Fuzzy Sets* karena metode ini dapat melakukan perhitungan terhadap nilai relevansi suatu halaman *web* dengan kata kunci yang di-*input*-kan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijabarkan sebelumnya, rumusan masalah dalam pembuatan tugas akhir ini adalah bagaimana membuat suatu perangkat lunak mesin pencari yang memiliki *web crawler* sebagai mesin yang bekerja mengunjungi situs-situs *web* dan mengumpulkannya ke dalam basis data sehingga dapat memfasilitasi pencarian informasi untuk publik serta menerapkan Metode *Conceptual Fuzzy Sets* dalam meranking hasil pencarian informasi yang dilakukan oleh perangkat lunak mesin pencari tersebut.

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah mesin pencari yang dibuat hanya dapat melakukan pencarian informasi terhadap halaman *web* saja, dan tidak menyediakan pencarian terhadap gambar atau *file-file* tertentu.

1.4 Tujuan Penelitian

Tujuan penulisan tugas akhir ini adalah membuat perangkat lunak mesin pencarian yang memiliki *web crawler* sebagai mesin yang bekerja mengunjungi situs-situs *web* dan mengumpulkannya ke dalam basis data serta memfasilitasi pencarian untuk publik dengan mengumpulkan hasil pencarian menggunakan Metode *Conceptual Fuzzy Sets*.

1.5 Sistematika Penulisan

Sistematika penulisan laporan ini disusun agar dalam pembuatan laporan dapat lebih terstruktur dan ringkas. Adapun urutan penulisan sebagai berikut :

BAB I : PENDAHULUAN

Bab I berisi penjelasan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan tugas akhir yang dibuat.

BAB II : LANDASAN TEORI

Bab II berisi uraian tentang defenisi dan teori pendukung serta teori penjelasan mesin pencarian (*Search Engine*) dengan metode yang digunakan dalam tugas akhir.

BAB III : METODOLOGI PENELITIAN

Bab III berisi tentang penelitian sebelumnya yang pernah dilakukan dan langkah-langkah penelitian yang dilakukan oleh peneliti selama pembuatan tugas akhir.

BAB IV : ANALISA DAN PERANCANGAN

Bab IV berisi pembahasan mengenai analisis kebutuhan sistem dan perancangan program yang akan dibuat.

BAB V : IMPLEMENTASI DAN PENGUJIAN

Bab V berisi penjelasan mengenai langkah-langkah pembuatan program dan hasil pengujian.

BAB VI : PENUTUP

Bab VI merupakan bagian akhir dari penulisan yang berisi kesimpulan dan saran yang diberikan penulis untuk pengembangan sistem.

BAB II

LANDASAN TEORI

Perangkat lunak adalah teknologi yang harus digunakan oleh semua orang yang akan membangun perangkat lunak, dengan melalui serangkaian proses menggunakan sekumpulan metode dan alat bantu (Pressman,1997). Rekayasa perangkat lunak dapat dibagi menjadi dua bagian yaitu ‘rekayasa’ dan ‘perangkat lunak’. Rekayasa adalah pemakaian ilmu pengetahuan untuk masalah praktis dari yang semula tidak ada menjadi ada. Sedangkan perangkat lunak adalah kumpulan program komputer dengan fungsi tertentu. Jadi perangkat lunak merupakan komponen sistem yang *logic* bukan fisik dan biasanya dibangun atau direkayasa tidak melalui proses manufaktur yang umum.

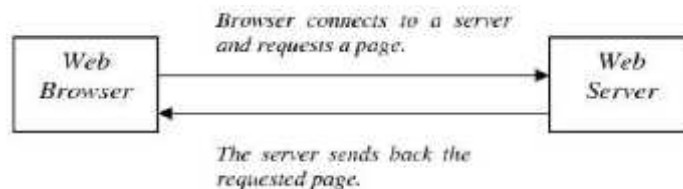
2.1 *Web Server*

Web server bisa diartikan sebagai sebuah komputer *server* yang memiliki kemampuan untuk memproses permintaan atau *request* dari *client* melalui protokol HTTP yang kemudian hasilnya akan dikirimkan kepada *client* yang terhubung dengan *web server* tersebut. *Reply* dari *server* tersebut akan ditampilkan pada komputer *client* melalui *web browser* yang ada di komputer *client*. *Web browser* adalah suatu program atau *software* yang didesain untuk mencari dan menampilkan informasi yang berupa *web pages* (Ricky Setiawan, 2005).

Protokol yang digunakan untuk melakukan proses *request* oleh *client* sampai dengan proses *reply* dari *server* yaitu protokol HTTP dengan standar *port*

yaitu *port* 80. Protokol ini merupakan singkatan dari *Hypertext Transfer Protocol*.

Berikut adalah gambar hubungan antara *web browser* dengan *web server*.



Gambar 2. 1 Hubungan antara *Web Browser* dan *Web Server* (Sumber : Ricky Setiawan, 2005)

Cara kerja *web server* pada umumnya adalah sebagai berikut. Misalkan *client* ingin mendapatkan informasi pada sebuah situs yang membahas tentang bagaimana cara kerja sebuah mesin pencari dan kebetulan *client* mengetahui *Uniform Resource Locator* (URL) dari situs itu, misal URL-nya adalah <http://www.searchenginewatch.com/index.html>, maka *client* mengetikkan URL dari situs tersebut pada *address bar* suatu *web browser* dan menekan tombol *enter*. Proses yang terjadi setelah itu adalah *web browser* memisahkan URL tadi menjadi 3 bagian, yaitu bagian *protocol* (http), bagian *server name* (www.searchenginewatch.com) dan bagian *file name* (index.html). Perlu diketahui bahwa proses ini tidak tampak oleh *client*. Setelah itu *web browser* mengubah bagian *server name* “www.searchenginewatch.com” ke bentuk *IP address* yang akan digunakan oleh *web browser* untuk melakukan koneksi dengan *server*. Setelah koneksi terbentuk *web browser* mengirim perintah *GET* kepada *server* untuk meminta *file* (index.html) yang diinginkan oleh *client*. Kemudian *server* memberikan suatu respon yaitu dengan mengirim *HTTP Header* dan *HTML*.

Content dari situs www.searchenginewatch.com/index.html kepada *web browse*.

Untuk HTTP *Header* dapat dilihat pada tabel 2.1. berikut ini :

Tabel Tabel 2.1 HTTP *Header*

HTTP Header	Keterangan
HTTP/1.1 200 OK	Menyatakan bahwa <i>request</i> telah berhasil diinterpretasikan atau diterjemahkan dan <i>file</i> yang di- <i>request</i> ada.
<i>Method: GET</i>	Metode yang digunakan untuk mengambil HTTP <i>Header</i> dan juga HTML <i>Content</i> -nya.
<i>Date: Thu, 17Mar 2005 06:05:06 GMT</i>	Menyatakan hari, tanggal, dan waktu pada saat dilakukan <i>request</i> .
<i>Server: Apache/2.0.40 (Win32)</i>	Menyatakan jenis <i>server</i> yang digunakan.
<i>Last-Modified: Mon, 28 Feb 2005 06:53:24 GMT</i>	Menyatakan hari, tanggal, dan waktu terakhir dilakukan modifikasi pada <i>file</i> yang di- <i>request</i> .
<i>Accept-Ranges: bytes</i>	Merupakan satuan yang dipakai untuk menyatakan ukuran <i>file</i> .
<i>Content-Length: 5743</i>	Menyatakan ukuran dari <i>file</i> .
<i>Content-Type: text/html;</i>	Menyatakan bahwa <i>file</i> yang di- <i>request</i> berupa <i>file</i> HTML.

Setelah itu *web browser* akan menterjemahkan (*render*) HTML *Content* yang berupa *file* atau dokumen HTML untuk kemudian ditampilkan di komputer *client* (Ricky Setiawan, 2005).

Pada Tabel 21. Diatas (pada kolom HTTP *Header*) terdapat informasi yang bertuliskan “HTTP/1.1 200 OK”, informasi “200 OK” itu menyatakan *Response Code* dan *Response Text* yang dikirim oleh *server* kepada *client*. *Response Text* merupakan keterangan dari *Response Code*. Untuk *Response Code* dan *Response Text* yang lain (hanya dituliskan beberapa saja) bisa dilihat pada pada Tabel 2.2 berikut ini :

Tabel Tabel 2. 2 *Response Code dan Response Text*

Response Code	Response Text
201	<i>Create</i>
202	<i>Accepted</i>
203	<i>Non-authoritative Information</i>

Tabel Tabel 2. 3 *Response Code dan Response Text (Lanjutan)*

<i>Response Code</i>	<i>Response Text</i>
204	<i>No Content</i>
205	<i>Reset Content</i>
301	<i>Moved Permanently</i>
302	<i>Moved Temporarily</i>
303	<i>See Other</i>
304	<i>Not Modified</i>
305	<i>Use Proxy</i>
400	<i>Bad Request</i>
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>
405	<i>Method Not Allowed</i>

2.2 Mesin Pencari (*Search Engine*)

Mesin pencari merupakan sebuah perangkat lunak atau program yang digunakan untuk mencari informasi berdasarkan kata kunci yang diberikan pengguna mesin pencari. Kata kunci tersebut akan diolah mesin pencari sampai pada akhirnya didapatkan informasi yang mempunyai relevansi dengan kata kunci tersebut. Informasi tersebut bisa berupa halaman-halaman *web (web pages)* ataupun gambar (*image*) (Ricky Setiawan, 2005).

Mesin pencari yang pertama kali muncul adalah "Wandex" dari *World Wide Web Wanderer*, suatu *web crawler* yang dibangun oleh Matthew Gray di *Massachusetts Institute of Technology* pada tahun 1993, namun sekarang wandex tidak berfungsi lagi. Pada tahun yang sama muncul mesin pencari bernama *Aliweb*, dan sampai saat ini masih beroperasi dengan baik. Pada tahun 1994, *Carnegie Mellon University* telah memperkenalkan sebuah mesin pencari pertama yang bernama *Lycos*, di mana mesin pencari ini sekarang berkembang menjadi usaha komersial yang cukup besar. Segera sesudah itu, beberapa mesin pencari lain banyak bermunculan dan bersaing untuk memperebutkan popularitas.

Termasuk diantaranya adalah *WebCrawler*, *Hotbot*, *Excite*, *Infoseek*, *Inktomi*, dan *AltaVista*. Mereka bersaing dengan direktori-direktori populer seperti misalnya Yahoo!(Fanny, 2008).

Perkembangan selanjutnya, direktori-direktori tersebut menggabungkan atau menambahkan teknologi mesin pencari agar fungsinya semakin meningkat seperti misalnya ZUULA, CUIL, dan FILESCOPE. Pada tahun 2002, Yahoo! Mengakuisisi Inktomi dan selang satu tahun mengakuisisi Overture yang memiliki *AlltheWeb* dan *Altavista*. Tahun 2004, Yahoo! meluncurkan mesin pencarinya sendiri yang didasarkan pada teknologi gabungan dari mesin-mesin yang telah diakuisisi tersebut dan memberikan pelayanan dengan mengutamakan mesin pencari dibandingkan direktorinya (Fanny, 2008).

Dalam konteks internet, mesin pencari biasanya merujuk kepada *www* dan bukan protokol ataupun area lainnya. Selain itu, mesin pencari mengumpulkan data yang tersedia di *newsgroup*, *database* besar, atau direktori terbuka seperti DMOZ.org. Karena pengumpulan datanya dilakukan secara otomatis, mesin pencari berbeda dengan direktori *web* yang dikerjakan manusia (Wikipedia, 2009).

Informasi yang diperoleh mesin pencari akan ditampilkan secaraurut berdasarkan nilai relevansinya. Informasi yang memiliki nilai relevansi paling tinggi berada pada halaman pertama pada posisi pertama (ranking pertama). Tiga komponen penting yang dimiliki oleh mesin pencari adalah sebagai berikut :

1. *Spider* atau *Crawler*

Spider atau *Crawler* adalah sebuah perangkat lunak atau program yang bertugas mengunjungi situs *web*, membaca isi dari situs *web*, dan mengikuti *link-link* yang ada pada situs *web* tersebut. *Crawler* akan mengunjungi situs *web* secara periodik untuk melihat apakah ada perubahan pada situs *web* tersebut.

2. *Index*

Dapat dianalogikan seperti katalog raksasa yang berisi *copy* dari setiap situs *web* yang dikunjungi oleh *crawler*. Jika *crawler* menemukan adanya perubahan pada situs *web* yang sebelumnya pernah dikunjungi maka informasi dari situs *web* tersebut (informasi yang ada dalam katalog) akan segera diperbarui atau di-*update*

3. *Software search engine*

Software atau program inilah yang akan memilah ribuan atau bahkan jutaan data/informasi yang ada dalam *index/katalog* tadi berdasarkan kata kunci yang diberikan oleh pengguna mesin pencari. Data/informasi yang dianggap mempunyai relevansi dengan kata kunci akan ditampilkan secaraurut sesuai dengan nilai relevansinya.

Setiap mesin pencari memiliki tiga komponen diatas, hanya saja berbeda dalam hal pengaturannya, seperti misalnya dalam hal strategi pencarian (*seacrh strategy*) yang akan digunakan, algoritma yang digunakan oleh mesin pencari, dan juga jumlah situs *web* yang di-*index* oleh *crawler*. Oleh karena itu apabila dilakukan perbandingan terhadap hasil dari dua buah mesin pencari dengan kata

kunci yang sama, akan didapatkan hasil yang berbeda, misalnya yaitu dalam hal jumlah total data/informasi yang didapat (data/informasi yang mempunyai relevansi dengan kata kunci) dan ranking dari data/informasi tersebut (Ricky Setiawan, 2005).

2.2.1 Cara Kerja Mesin Pencari

Mesin pencari dirancang untuk meng-*index* hasil temuan berupa alamat *web* (URL, FTP, dan lain-lain) dari sistem *database*. Untuk menunjang tugas tersebut, mesin pencari dilengkapi program khusus yaitu: *spider*, bot (robot), dan *crawler*. *Spider/crawler* dan robot akan “merayap” dari satu alamat *web* ke alamat *web* lain, dari satu *link* ke *link* lainnya. Ketika *spider/crawler* tersebut tiba pada sebuah halaman *web*, maka *spider/crawler* akan membuat *index* dari sebagian besar kata-kata yang terdapat dalam situs tersebut. Sedangkan pemilik halaman *web* mengirimkan URL situsnya ke mesin pencari agar dapat ditelusuri oleh *spider/crawler* dan dicantumkan ke dalam *database* mesin pencari tersebut. Ketika pengguna mesin pencari mencari data/informasi di *web* menggunakan mesin pencari, sebenarnya pengguna mesin pencari membuat permintaan kepada mesin pencari tersebut untuk mengecek *index* yang telah disusun oleh mesin pencari pada *database*-nya, kemudian mencocokkannya dengan kata kunci yang diminta. Setelah data sesuai dengan yang diminta, mesin pencari akan menyajikan kepada pengguna dalam bentuk daftar *link* pada layar komputer, selanjutnya informasi tersebut dapat di-*click* pengguna bisa secara langsung terhubung ke situs sesuai tujuan.

Data tentang halaman *web* disimpan dalam sebuah *database index* untuk digunakan dalam pencarian selanjutnya. Sebagian mesin pencari, seperti Google (maupun mesin pencari yang sejenis), menyimpan seluruh maupun sebagian halaman sumber (*cache*) atau informasi tentang halaman *web* itu sendiri. Ada jenis mesin pencari lain: mesin pencari *real-time*, seperti *Orase*. Mesin pencari jenis ini tidak menggunakan indeks. Informasi yang diperlukan oleh mesin tersebut hanya dikumpulkan jika ada pencarian baru. Jika dibandingkan dengan sistem berbasis indeks yang digunakan mesin-mesin seperti Google, sistem *real-time* ini unggul dalam hal: informasi selalu mutakhir, dan lebih sedikit sumber daya sistem yang diperlukan. (Wikipedia, 2009)

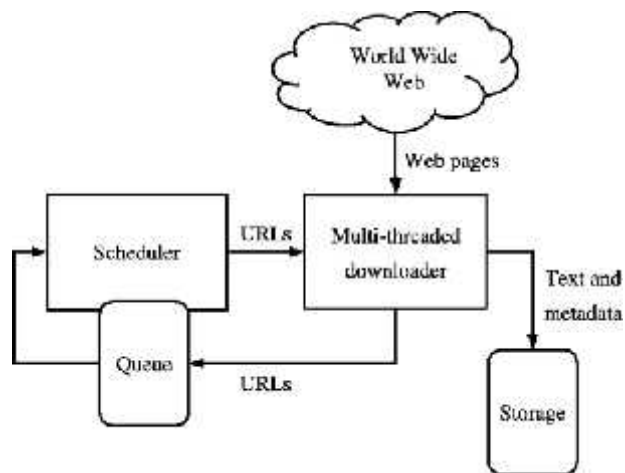
2.3 *Web Crawler*

Web crawler adalah sebuah perangkat lunak atau program yang bertugas mengunjungi situs *web*, membaca isi dari situs *web*, dan mengikuti *link-link* yang ada pada situs *web* tersebut. *Crawler* akan mengunjungi situs *web* secara periodik untuk melihat apakah ada perubahan pada situs *web* tersebut (Ricky Setiawan, 2005). *Web Crawler*, berfungsi sebagai *web spider* atau *web robot* di dalam mesin pencari. Hasil pengumpulan situs *web* selanjutnya akan di-*index* oleh mesin pencari dan menyimpan nya kedalam *database* (Dwi H Widyantoro, 2006).

2.3.1 *Cara Kerja Web Crawler*

Proses sebuah *Web Crawler* pada umumnya dimulai dengan memberikan sekumpulan awal URL sebagai benih penelusuran ke dalam sebuah antrian. Kriteria prioritas dapat diterapkan untuk menyusun ulang daftar URL pada antrian tersebut. *Crawler* selanjutnya men-*download* halaman *Web* berdasarkan URL

yang diambil dari antrian. Setelah disimpan ke dalam koleksi, halaman *web* yang diperoleh diurai (*parsed*) untuk diekstrak *out-going link* yang belum dikunjungi dan selanjutnya dimasukkan ke dalam antrian. Proses jemputan halaman *web* berlangsung terus sampai antrian URL kosong atau kalau kondisi berhenti sudah terpenuhi (Dwi H Widyantoro, 2006).



Gambar 2. 2 Langkah-langkah kerja *Web Crawler* (Sumber : bakauheni.com 2009)

Untuk situs *web* yang memiliki frekuensi *update* halaman yang tinggi maka diberlakukan 2 aturan:

1. ***Uniform Policy*** adalah *Re-visiting* semua halaman *web* dengan frekuensi yang sama.
2. ***Proportional Policy*** adalah Semakin sering halaman *web* tersebut *update* maka semakin sering juga halaman tersebut dikunjungi oleh *crawler*

Akan tetapi untuk halaman *web* yang sangat terlalu sering di-*update* maka *Proportional Policy* tidak berlaku (Adith, 2007).

2.4 *Fuzzy*

Fuzzy (samar) merupakan suatu teori tentang konsep penilaian, dan segala sesuatu merupakan persoalan derajat atau diibaratkan bahwa segala sesuatu memiliki elastisitas.

Fuzzy ditemukan oleh Prof. Lotfi Zadeh dari Universitas California pada tahun 1960. Prinsip *fuzzy* ditemukan atas dasar adanya ketidakakuratan dalam perhitungan logika *boolean* yang tidak memperhatikan kondisi-kondisi terkait pada permasalahan. Karena kekurangakuratan perhitungan pada masalah tidak terbatas jumlah benar dan salahnya zadeh mengembangkan ide penggolongan set yang dinamakan *fuzzy set*. *Fuzzy set* dalam perhitungannya, banyak memiliki nilai kondisi. *Fuzzy* membaginya dalam derajat keanggotaan dan derajat kebenaran, yaitu : sesuatu yang dapat menjadi sebagian benar dan sebagian salah pada waktu yang sama.

2.4.1 Variabel Linguistik

Variabel linguistik adalah variabel yang bukan bernilai angka melainkan bernilai kata/kalimat. Sebagai alasan menggunakan kata/kalimat dari pada angka karena peranan linguistik kurang spesifik dibandingkan angka, namun informasi yang disampaikan lebih informatif. Variabel linguistik ini merupakan konsep penting dalam logika *fuzzy* dan memegang peranan penting dalam beberapa aplikasi.

Zadeh memberi konsep tentang variabel linguistik yang dikarakteristikan sebagai berikut :

$$(X, T(x), U, G, M)$$

Keterangan :

X = nama variabel (variabel linguistik)

T(x) = semesta pembicaraan untuk x atau disebut juga nilai linguistik dari x

U = jangkauan dari setiap nilai samar untuk x yang dihubungkan dengan variabel dasar U

G = aturan sintaksis untuk memberikan nama (x) pada setiap nilai X

M = aturan semantik yang menghubungkan setiap X dengan artinya.

Contoh :

X = “kecepatan”, dengan $U[0, 100]$ dan $T(\text{kecepatan}) = \{\text{lambat, sedang, cepat}\}$

Maka M untuk setiap X, M(x) adalah M(lambat), M(sedang), M(cepat). Dimana :

M(lambat) = himpunan samarnya “kecepatan dibawah 40 mph” dengan fungsi keanggotaan μ lambat.

M(sedang) = himpunan samarnya “kecepatan mendekati 55 mph” dengan fungsi keanggotaan μ sedang.

M(cepat) = himpunan samarnya “kecepatan diatas 70 mph” dengan fungsi keanggotaan μ cepat.

2.4.2 Fungsi keanggotaan

Mendefinisikan keanggotaan himpunan *fuzzy* dapat dilakukan dengan dua cara, yaitu secara numeris dan fungsional. Definisi numeris menyatakan fungsi derajat keanggotaan sebagai vektor jumlah yang tergantung pada tingkat diskretisasi. Misalnya, jumlah elemen diskret dalam semesta pembicaraan. Definisi fungsional menyatakan derajat keanggotaan sebagai batasan ekspresi

analitis yang dapat dihitung. Standar atau ukuran tertentu pada fungsi keanggotaan secara umum berdasar atas semesta X bilangan real.

2.4.3 *Conceptual Fuzzy Sets*

Conceptual Fuzzy Sets Theory merupakan pengembangan dari *Fuzzy Sets Theory* yang digunakan untuk mengatasi kekurangan yang ada pada *Fuzzy Sets Theory*. *Conceptual fuzzy sets* berkerja dengan cara mengekspansi kata kunci / kata kunci frasa utama menjadi beberapa kata kunci / kata kunci frasa lain dengan memberi nilai untuk tiap kata kunci / kata kunci frasa. kata kunci / kata kunci frasa yang memiliki nilai paling tinggi berarti memiliki nilai relevansi yang paling kuat dengan kata kunci / kata kunci frasa utama.

Conceptual Fuzzy Sets mampu menyajikan informasi yang tepat meskipun kata kunci / kata kunci frasa yang di-*input*-kan berupa suatu konteks kalimat. Di dalam menyajikan informasi, *Conceptual Fuzzy Sets Theory* menggunakan dua rumus yang merupakan standar *fuzzy*. Rumus yang digunakan adalah sebagai berikut (Ricky Setiawan, 2005) :

1. Rumus relasi “or” (*join* atau *union*) dalam *fuzzy* menggunakan rumus

$$a \cup b = \max (a , b)$$

2. Rumus relasi “and” (*intersection*) dalam *fuzzy* menggunakan rumus

$$a \cap b = \min (a , b)$$

Pencarian dengan menggunakan *conceptual fuzzy sets* tidak hanya memberikan suatu informasi yang memiliki kata kunci yang diberikan. Bisa juga suatu informasi yang tidak mengandung *keyword* yang di-*input*-kan tetapi memiliki relevansi kata kunci. Hal tersebut dapat terjadi karena dalam melakukan

pencarian *conceptual fuzzy sets* menggunakan *extended keyword*. *Extended keyword* adalah kata kunci - kata kunci lain yang memiliki relevansi dengan kata kunci utama meskipun secara sintaks berbeda.

Extended keyword diperoleh melalui proses "*Maintenance Keyword*" menggunakan perhitungan "*Conditional Probability Relation*" dengan sifat *Asymmetric*. Perhitungan ini menggunakan rumus yaitu:

$$P(k_1 | k_2) = \frac{P(k_1 \cap k_2)}{(k_2)} \dots\dots\dots(2.1)$$

$$P(k_2 | k_1) = \frac{P(k_1 \cap k_2)}{(k_1)} \dots\dots\dots(2.2)$$

$$P(k_1 \cap k_2) = \frac{\text{jumlah web page yang mengandung keyword } k_1 \text{ dan } k_2}{\text{Jumlah total web page dalam database}} \dots\dots\dots(2.3)$$

$$P(k_1) = \frac{\text{jumlah web page yang mengandung keyword } k_1}{\text{Jumlah total web page dalam database}} \dots\dots\dots(2.4)$$

$$P(k_2) = \frac{\text{jumlah web page yang mengandung keyword } k_2}{\text{Jumlah total web page dalam database}} \dots\dots\dots(2.5)$$

Keterangan :

$P = Probability$

$k_1 = Keyword 1$

$k_2 = Keyword 2$

Contoh penerapan rumus *Conditional Probability Relation* dalam pencarian :

Misalkan kata kunci utama yang ada pada tabel adalah “sistem” dan “inventori”. Dalam *database* terdapat 10 halaman *web* dimana di dalam tiap dokumen terdapat hampir seluruhnya ada kata “sistem” dan “inventori”. Berikut gambaran sederhana *database* :

Tabel 2. 4 Gambaran Sederhana *Database*

Halaman Web	Keyword
1	sistem dan inventori
2	Sistem
3	sistem dan inventori
4	Inventori
5	Sistem
6	Sistem
7	Inventori
8	sistem dan inventori
9	---
10	---

Langkah 1 :

Cari *probability* dari semua kata kunci utama, dalam hal ini yaitu $P(\text{sistem})$ dan $P(\text{inventori})$. Perhitungan sebagai berikut :

$$P(\text{sistem}) = \frac{6}{10} = 0.6 \text{ (ada 6 dari 10 web page yang memiliki kata “sistem”)}$$

$$P(\text{inventori}) = \frac{5}{10} = 0.5 \text{ (ada 5 dari 10 web page memiliki kata “inventori”)}$$

Langkah 2 :

Cari *probability* antar kata kunci utama, yaitu $P(\text{sistem} \cap \text{inventori})$.

Perhitungan sebagai berikut :

$$P(\text{sistem} \cap \text{inventori}) = \frac{3}{10} = 0.3 \text{ (ada 3 dari 10 web page)}$$

Langkah 3 :

Cari nilai relevansi antar kata kunci

$$P(\text{sistem} \mid \text{sistem inventori}) = \frac{P(\text{sistem} \cap \text{inventori})}{P(\text{sistem})} = \frac{0.3}{0.5} = 0.6$$

$$P(\text{inventori} \mid \text{sistem informasi}) = \frac{P(\text{sistem} \cap \text{inventori})}{P(\text{sistem})} = \frac{0.3}{0.6} = 0.5$$

Jika kata kunci “sistem” diasumsikan sebagai *fuzzy sets* terhadap kata kunci lainnya, maka hasilnya dapat ditulis sebagai berikut :

$$\text{sistem} = (1.0/\text{sistem}, 0.6/\text{inventori})$$

Begitu juga jika kata kunci “inventori” diasumsikan sebagai suatu *fuzzy sets* terhadap kata kunci lainnya, maka hasilnya dapat ditulis sebagai berikut :

$$\text{inventori} = (1.0/\text{inventori}, 0.5/\text{sistem})$$

kata “sistem” (di sebelah kiri “=”) dianggap sebagai *fuzzy label*, sedangkan kata “sistem”, “inventori” (di sebelah kanan “=”) dianggap sebagai elemen yang merupakan *extended keyword*.

Nilai 1.0, 0.6 menyatakan nilai relevansi antara elemen yang bersesuaian dengan *fuzzy label*nya. Sebagai contoh , kata “inventori” memiliki nilai relevansi 0.6 terhadap kata “sistem”.

2.5 *Term Frequency – Inverse Document Frequency (TF-IDF) Ranking*

TF-IDF *Ranking* merupakan metode atau cara yang digunakan untuk menghitung bobot sebuah kata/*term* terhadap sebuah halaman *web*, maksudnya yaitu apakah kata/*term* tersebut memiliki arti atau peranan yang penting dalam

sebuah halaman *web*. Untuk menentukan bobot tersebut ditentukan oleh dua faktor, yaitu *Term Frequency* (TF) dan *inverse document frequency* (IDF).

Term frequency adalah jumlah kemunculan kata/*term* dalam halaman *web*. Sedangkan *document frequency* adalah jumlah halaman *web* yang mengandung kata/*term* tersebut. Bobot dari sebuah kata/*term* akan tinggi apabila nilai *term frequency*-nya tinggi tetapi nilai *document frequency*-nya rendah, namun jika nilai *term frequency* dan *document frequency* sama-sama tinggi maka bobot dari kata/*term* tersebut akan rendah, sehingga kata/*term* tersebut dapat dianggap sebagai *stopwords* atau bisa juga merupakan kata/*term* yang sering digunakan dalam kehidupan sehari-hari.

Berikut adalah rumus yang digunakan untuk menghitung bobot dari sebuah kata/*term* :

$$w_{ij} = t_{ij} \times \frac{N \times t_{ij}}{\sum_{k=1}^N t_{kj}} \dots\dots\dots(2.6)$$

$$t_{ij} = 0.5 + 0.5 \times \frac{t_{ij}}{\max_{k=1}^N (t_{kj})} \dots\dots\dots(2.7)$$

Keterangan :

- w_{ij} = bobot kata/*term* t_j terhadap halaman *web* p_i
- t_{ij} = jumlah kemunculan kata/*term* t_j dalam halaman *web* p_i
- t_{kj} = jumlah kemunculan kata/*term* t_j pada keseluruhan halaman *web*
- N = jumlah semua halaman *web* yang ada dalam *database*

2.6 *Regular Expression*

Regular expression atau yang sering disebut sebagai *Regex* adalah sebuah formula untuk pencarian pola suatu kalimat/string (Agus Muliantara,2009). Sering kali orang beranggapan bahwa *regex* susah dan membingungkan. Namun sebenarnya *regex* sangatlah membantu dalam menemukan pola-pola kalimat. Sehingga percobaan terhadap semua kemungkinan pola kalimat tidak perlu dilakukan.

Regular expression umumnya digunakan oleh banyak pengolah kata/*text* editor dan peralatan lainnya untuk mencari dan memanipulasi kalimat dengan berdasarkan kepada suatu pola tertentu.

Banyak bahasa pemrograman yang mendukung *regular expression* seperti misalnya PHP, perl, VB dan Tcl. Sebuah alasan yang sangat bagus untuk menggunakan *regex* adalah karena *regex* sangatlah powerfull. Pada level rendah *regex* dapat mencari sebuah penggalan kata. Pada level tinggi *regex* mampu melakukan kontrol terhadap data. Baik mencari, menghapus dan merubah.

Mari kita pikirkan bagaimana cara untuk mencari sebuah file di hard disk. Seringkali digunakan karakter “?” dan “*”. Penggunaan karakter “?” mengandung arti bahwa sedang dicari sebuah file yang mengandung sebuah karakter tertentu dan karakter “*” mengandung arti sedang dicari nol atau lebih karakter. Sebagai contoh : pencarian dengan menggunakan pola “file?.dat” akan menghasilkan beberapa contoh sebagai berikut :

1. file1.dat
2. file2.dat

3. file3.dat
4. file.dat
5. fileN.dat

Menggunakan karakter “*” akan menghasilkan lebih banyak hasil pencarian. “file*.dat” akan menghasilkan :

1. file1.dat
2. file2.dat
3. file12.dat
4. filex.dat
5. fileXYZ.dat

Kedua contoh di atas adalah salah satu penggunaan *regular expression*. Bayangkan saja jika pencarian *file* hanya menggunakan metode nama *file* tanpa menyertakan *regular expression*. Tentunya saat proses pencarian menggunakan pola “file*.dat” hasilnya adalah harus “file*.dat”. Hal inilah yang menyebabkan *regex* menjadi sangat *powerfull*. Dengan menggunakan sebuah pola didapatkan hasil yang luas.

Beberapa pola yang umum digunakan dalam *regex* tampak pada tabel 1 berikut.

Tabel 2. 5 Pola Umum *Regex*

Pola	Penjelasan
[]	Ekspresi kurung. cocok dengan satu karakter yang berada dalam kurung, misal: pattern "a[bcd]i" cocok dengan string abi, "aci", dan "adi". penggunaan range huruf dalam kurung diperbolehkan, misal : pattern "[a-z]" cocok dengan salah satu karakter diantara string "a" sampai "z". pattern [0-9] cocok dengan salah satu angka. jika ingin mencari karakter "-" juga, karakter tersebut harus diletakkan di depan atau di belakang kelompok, misal: "[abc-]".

Tabel 2. 6 Pola Umum *Regex* (Lanjutan)

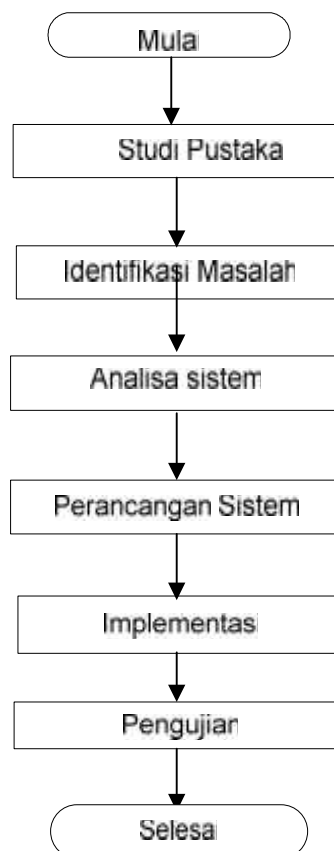
Pola	Penjelasan
[]	Ekspresi kurung. cocok dengan satu karakter yang berada dalam kurung, misal: pattern "a[bcd]i" cocok dengan string abi, "aci", dan "adi". penggunaan range huruf dalam kurung diperbolehkan, misal : pattern "[a-z]" cocok dengan salah satu karakter diantara string "a" sampai "z". pattern [0-9] cocok dengan salah satu angka. jika ingin mencari karakter "-" juga, karakter tersebut harus diletakkan di depan atau di belakang kelompok, misal: "[abc-]".
[^]	Cocok dengan sebuah karakter yang tidak ada dalam kurung, berlawanan dengan yang diatas. misal: pattern "[^abc]" cocok dengan satu karakter apa saja kecuali a, b, "c".
?	Cocok dengan nol atau satu karakter sebelumnya. misal: pattern "died?" cocok dengan string "die" dan "died".
+	Cocok dengan satu atau lebih karakter sebelumnya. misal: "yu+k" cocok dengan yuk, "yuuk", "yuuuk", dan seterusnya.
*	Cocok dengan nol atau lebih karakter sebelumnya. misal: pattern "hu*p" cocok dengan string "hp", "hup", "huup" dan seterusnya.
{ x }	Cocok dengan karakter sebelumnya sejumlah x karakter. misal: pattern "[0-9]{3}" cocok dengan bilangan berapa saja yang berukuran 3 digit.
{ x,y }	Cocok dengan karakter sebelumnya sejumlah x hingga y karakter. misal: pattern "[a-z]{3,5}" cocok dengan semua susunan huruf kecil yang terdiri dari 3 sampai 5 huruf.
!	Jika diletakkan di depan pattern, maka berarti "bukan". misal pattern "!a.u" cocok dengan string apa saja kecuali string "alu", anu, "abu", "asu", "aiu", dan seterusnya.
^	Jika diletakkan di depan pattern, akan cocok dengan awal sebuah string.
\$	Jika diletakkan di belakang pattern, akan cocok dengan akhir sebuah string
()	Gruping. digunakan untuk mengelompokkan karakter-karakter menjadi single unit. string yang cocok dalam pattern yang berada dalam tanda kurung dapat digunakan pada operasi berikutnya. semacam variable.
\	Escape character. mengembalikan fungsi metacharacter menjadi karakter biasa. pada beberapa system dapat berarti sebaliknya, yaitu metacharacter menggunakan escape character didepannya

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian menguraikan seluruh kegiatan yang dilaksanakan selama kegiatan penelitian berlangsung. Adapun langkah-langkah yang dilalui dalam pelaksanaan penelitian ini adalah sebagai berikut:



Gambar 3.1 Bagan Metodologi Penelitian

3.1.1 Studi Pustaka

Tahap ini adalah tahap awal melakukan penelitian tugas akhir ini. Pada tahap ini, dilakukan pencarian informasi mengenai penelitian-penelitian yang pernah dilaksanakan sebelumnya yang berhubungan dengan pembuatan perangkat lunak mesin pencari. Informasi-informasi ini akan digunakan untuk mengidentifikasi masalah. Pencarian informasi dilakukan dengan melaksanakan studi literatur yang bertujuan untuk memperoleh teori-teori dan konsep-konsep yang mendasar mengenai materi yang berhubungan dalam pembuatan perangkat lunak mesin pencari, yaitu dengan cara mempelajari buku-buku, artikel-artikel, jurnal-jurnal dan media lainnya.

3.1.2 Identifikasi Masalah

Dengan memanfaatkan informasi-informasi yang didapat dari persiapan penelitian dan studi literatur yang telah dilakukan, maka dilakukan tahap berikutnya yaitu mengidentifikasi masalah. Pada tugas akhir ini masalah yang akan diidentifikasi adalah bagaimana membuat suatu perangkat lunak mesin pencari yang memiliki *web crawler* sebagai mesin yang bekerja mengunjungi situs-situs *web* dan mengumpulkannya ke dalam basis data sehingga dapat memfasilitasi pencarian informasi untuk publik. Menerapkan metode *Conceptual Fuzzy Sets* dalam merankingkan hasil pencarian informasi yang dilakukan oleh perangkat lunak mesin pencari tersebut.

3.1.3 Analisa Sistem

Pada tahap ini peneliti melakukan analisa kebutuhan sistem yang akan dibuat. Analisa yang dibuat adalah analisa sistem lama dan analisa sistem baru. Pada analisa sistem lama, mesin pencari Google akan dijadikan sebagai bahan analisa. Sedangkan analisa sistem baru mencakup kebutuhan data, analisa metode *conceptual fuzzy sets*, dan fungsional. Submenu kebutuhan data membahas mengenai proses *input* data sampai dengan proses *output* data, submenu analisa metode *conceptual fuzzy sets* membahas mengenai tahapan / proses dalam metode *conceptual fuzzy sets*, sedangkan submenu fungsional membahas mengenai *Context Diagram* dan Diagram Alir Data (*Data Flow Diagram*).

3.1.4 Perancangan Sistem

Pada tahap ini, dilakukan perancangan terhadap sistem yang akan dibangun. Perancangan sistem meliputi perancangan basis data, perancangan struktur menu dan perancangan *interface*.

3.1.5 Implementasi

Tahap Implementasi merupakan tahap dimana sistem siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan aplikasi yang sesuai dengan tujuan penelitian yaitu berupa perangkat lunak mesin pencari yang memiliki *web crawler* sebagai mesin yang bekerja mengunjungi situs-situs *web* dan mengumpulkannya ke dalam basis data sehingga dapat memfasilitasi pencarian informasi untuk public

serta menerapkan Metode *Conceptual Fuzzy Sets* dalam merankingkan hasil pencarian informasi yang dilakukan oleh perangkat lunak mesin pencari tersebut.

3.1.6 Pengujian

Tahap selanjutnya yaitu melakukan pengujian terhadap perangkat lunak yang telah dibangun. Jika terdapat kesalahan seperti tidak sesuai hasil pencarian dengan perhitungan manual metode *conceptual fuzzy sets*, atau tidak berjalan sebagaimana analisa sistem yang dijabarkan dalam laporan, maka proses akan kembali ke tahap *design* sistem yang terdiri dari tahapan analisa sistem, perancangan sistem dan implementasi untuk dilakukan pengecekan ulang.

BAB IV

ANALISA DAN PERANCANGAN

Analisa sistem dilakukan setelah tahap perencanaan dan sebelum tahap perancangan sistem. Kesalahan pada tahap ini menyebabkan kesalahan pada tahap selanjutnya. Proses analisis dilakukan untuk pemeriksaan masalah dan penyusunan alternatif pemecahan masalah yang timbul serta membuat spesifikasi sistem yang baru atau sistem yang akan diusulkan dan dimodifikasi.

4.1 Analisa Sistem

Analisa sistem yang digunakan adalah dengan menggunakan analisa sistem yang telah ada, analisa sistem yang akan dibuat, Diagram Konteks, dan Diagram Alir Data.

4.1.1 Analisa Sistem Lama

Analisa sistem lama dilakukan untuk mendapatkan informasi penting dan menjadi masukan bagi sistem baru yang akan dikembangkan agar mampu memberikan hasil yang lebih baik dari sistem yang telah ada. Pada analisa sistem lama ini, Google atau yang lebih dikenal dengan URL www.google.com akan dijadikan sebagai bahan analisa. Pengambilan Google sebagai bahan analisa dikarenakan Google pada saat ini menjadi salah satu mesin pencari tercanggih dalam pencarian diinternet. Hal ini dijelaskan dengan ditempatkannya Google dirating teratas dari mesin pencari lainnya seperti, wisenut.com, alltheweb.com, altavista.com, dan lain-lain. Beberapa alasan mengapa Google adalah yang terbaik adalah:

1. *Crawler* Google.com terbukti mampu mengindeks situs dan informasi dari *server* ataupun *database* yang tersembunyi dibalik *firewall* atau yang dikhususkan sebagai informasi berbayar.
2. Banyaknya informasi atau situs yang ditampung oleh *database* Google.com adalah yang terbanyak diantara mesin pencari lainnya. Berdasarkan riset yang dilakukan oleh situs Searchenginesshowdown.com terdapat sekitar 968 juta buah informasi dan situs yang telah disimpan ke *database* Google.com.
3. Fasilitas pemilihan jenis bahasa sebagai antarmukanya lebih fleksibel. Ada 74 bahasa yang bisa dipilih, termasuk bahasa Indonesia, Jawa, Urdu, Klingon, dan sebagainya.
4. Google.com adalah satu-satunya mesin pencari umum yang menyimpan informasi situs ketika pertama kali mereka diindeks, biasa disebut sebagai *cached pages*. Sehingga meskipun suatu situs telah mati tetapi masih terdapat *link*-nya di dalam *database* Google.com, maka tampilan atau isi situs tersebut masih bisa dilihat dan dibaca.
5. Selain mampu mencari informasi di situs internet, Google.com juga mampu membongkar isi *file* yang berakhiran pdf, doc, xls, ppt, rtf, ps, txt, wpd, asp dan lainnya.

4.1.2 Analisa Sistem Baru

Sistem yang akan dibuat berupa perangkat lunak yang menyediakan proses pencarian informasi. Sistem ini dibantu oleh *crawler* dalam membangun *database*

dan mencari data dari halaman-halaman *web* di berbagai situs berita. Berbeda dengan *crawler* lainnya, *crawler* pada mesin pencari ini hanya akan mengunjungi, membaca serta mengumpulkan dan menyimpan halaman *web* pada situs *web* yang telah terdaftar dalam data master pada *database* perangkat lunak ini. Meskipun dalam suatu halaman *web* tersebut terdapat *link-link* yang mengarahkan *crawler* ke halaman *web* lainnya yang terdapat pada situs *web* yang berbeda, maka *crawler* tersebut tidak akan memngunjungi atau menyimpannya ke dalam *database*.

Fasilitas pencarian pada perangkat lunak ini akan dibangun menggunakan metode *conceptual fuzzy set*. Metode tersebut akan membantu mesin pencari dalam meranking hasil pencarian yang dilakukan publik (*user* umum) dalam melakukan pencarian informasi. Dimana halaman *web* yang dicari disesuaikan dengan kata kunci yang dimasukkan publik untuk mencari informasi yang diinginkan dalam mesin pencari ini, serta mesin pencari akan menampilkan hasil pencarian berdasarkan nilai hasil perhitungan metode secara terurut dari nilai terbesar hingga terkecil.

4.1.2.1 Kebutuhan Data

4.1.2.1.1 Analisa Data Inputan

Analisa data inputan bertujuan menganalisis data-data inputan yang berguna dalam proses kerja aplikasi mesin pencari dan *crawler*-nya. Data inputan yang berguna untuk sistem yaitu :

1. Data *URL*, berisi informasi *URL* utama sebagai tujuan awal *crawler* dalam mengumpulkan data. Data meliputi kode *URL* dan *URL*.

2. Data *crawler* untuk proses pencarian, berisi data yang dibangun oleh hasil *crawling* yang bertujuan sebagai data untuk proses pencarian.

4.1.2.1.2 Analisa Data Keluaran

Keluaran dari sistem yang dibangun adalah berupa daftar *URL* halaman *web* yang telah dipilih berdasarkan kata kunci yang dimasukkan publik dalam mencari informasi, serta data-data hasil perhitungan pencarian menggunakan Metode *Conceptual Fuzzy Sets*. Hasil pencarian akan ditampilkan secara berurut berdasarkan nilai terbesar hingga terkecil dari perhitungan dilakukan terhadap halaman *web* yang memiliki kata kunci sesuai dengan yang dicari oleh publik.

4.1.2.1.3 Analisa Fungsi Sistem

Perangkat lunak mesin pencari dengan dua golongan pengguna memiliki dua fungsi yang berbeda. Perbedaan fungsi disesuaikan dengan penggunanya, yaitu analisa fungsi sistem untuk Administrator dan analisa fungsi sistem untuk Publik.

Sistem yang dibangun untuk Administrator memiliki fungsi sebagai berikut :

1. Mengelola data master *URL* untuk digunakan oleh *crawler* dalam mengumpulkan halaman *web* dan membangun data dalam *database* untuk fasilitas pencarian dalam perangkat lunak mesin pencari
2. Memulai atau menjalankan *crawler* untuk menelusuri halaman *web*

Sedangkan sistem yang dapat diakses oleh publik memiliki fungsi sebagai berikut :

1. Pencarian informasi berupa halaman *web* berdasarkan kata kunci yang dimasukkan.
2. Meranking dan mengurutkan berdasarkan nilai terbesar untuk hasil pencarian berdasarkan nilai hasil perhitungan menggunakan metode *Conceptual Fuzzy Sets* yang telah dihitung secara otomatis oleh sistem.
3. Memanggil halaman *web* yang dipilih publik dengan mengklik *URL* yang dianggap sebagai hasil pencarian yang tepat.

4.1.2.2 Analisa *Conceptual Fuzzy Sets* Dalam *Search Engine*

Dalam pencarian informasi, mesin pencari ini didukung oleh Metode *Conceptual Fuzzy Sets* yang bertujuan untuk membantu perhitungan nilai suatu halaman *web* yang memiliki kata kunci yang dicari. Dan menentukan urutan ranking halaman *web* dalam pencarian. Langkah yang digunakan dalam perhitungan menggunakan metode *conceptual fuzzy sets* adalah sebagai berikut :

4.1.2.2.1 Input Kata Kunci

Kata kunci yang dimasukkan adalah kata kunci umum, artinya aplikasi tidak akan memperhatikan simbol-simbol yang terdapat pada kata kunci yang seperti bisa dilakukan pada Google, contohnya pencarian frase pada Google, yaitu pencarian yang hanya menghasilkan informasi sesuai dengan kalimat yang dimasukkan pada *text* kata kunci. Di Google, pencarian ini dapat dilakukan dengan menambahkan tanda kutip (“”) pada awal dan akhir kalimat.

Misalkan kata kunci = Gayus Tambunan

Berdasarkan kinerja utama metode *conceptual fuzzy sets* yang berkerja mengexpansi kata kunci utama yang dalam konteks kalimat menjadi bebarapa

kata kunci tambahan (*extended keyword*) berupa kata yang memiliki kesamaan secara makna dan konsep meskipun tidak memiliki kesamaan kalimat dengan kata kunci utama. Maka dengan metode ini dapat diperoleh kata kunci tambahan yang dinyatakan dengan keterangan sebagai berikut :

a = Gayus

b = Tambunan

sehingga jika digunakan rumus utama dalam metode *conceptual fuzzy sets* yaitu :

1. Rumus relasi “or” (*join* atau *union*)

$$a \cup b = \max(a, b)$$

2. Rumus relasi “and” (*join* atau *union*)

$$a \cap b = \min(a, b)$$

Diperoleh hasil :

1. $\text{Gayus} \cup \text{Tambunan} = \max(\text{Gayus}, \text{Tambunan})$

2. $\text{Gayus} \cap \text{Tambunan} = \min(\text{Gayus}, \text{Tambunan})$

Dari rumus

1. Max (Gayus , Tambunan)

Maka diperoleh hasil kalimat Gayus Tambunan. Sehingga dalam pencarian kalimat di dalam *web* yaitu kalimat atau frasa Gayus Tambunan. Kalimat yang berbeda meskipun memiliki kemiripan dalam beberapa kata maka tidak dinyatakan sebagai kata kunci.

2. Min (Gayus , Tambunan)

Hasil yang akan diperoleh yaitu kalimat yang memiliki kesamaan dalam beberapa kata meskipun berbeda dalam kesamaan kalimat. Maka dinyatakan dalam kesamaan makna. Kalimat yang mungkin akan muncul dengan kesamaan kata yang dapat dinyatakan sebagai kata kunci tambahan yaitu : gayus, saudara gayus, mafia gayus, gayus halomoan tambunan, tambunan, dan lainnya yang memiliki kata gayus atau tambunan salah satunya.

Pada fasilitas pencarian yang tersedia dalam perangkat lunak mesin pencari ini. Mesin pencari akan melakukan pencarian kalimat sesuai dengan hasil dari rumus diatas yaitu :

1. Halaman web yang memiliki kalimat “Gayus Tambunan”
2. Halaman web yang memiliki kalimat yang terdapat kata Gayus atau kata Tambunan

4.1.2.2 Proses Perhitungan Relevansi

Perhitungan nilai kata kunci dilakukan dengan menggunakan perhitungan *Conditional Probability Relation*. Perhitungan ini menggunakan rumus persamaan 2.1 sampai dengan persamaan 2.5 pada bab 2.

Tabel 4. 1 Jumlah Halaman *Web*

Keterangan	Jumlah halaman <i>web</i>
Gayus Tambunan	2
Gayus	7
Tambunan	3
Total <i>Web</i> Yang Memiliki Kata Kunci	7
Total <i>Web</i> Dalam <i>Database</i>	13

Langkah 1 :

Cari *probability* dari semua kata kunci utama, dalam hal ini yaitu $P(\text{Gayus})$ dan $P(\text{Tambunan})$. Perhitungan sebagai berikut :

$$P(\text{Gayus}) = \frac{7}{13} = 0.538 \text{ (ada 7 halaman } web \text{ yang memiliki kata "Gayus")}$$

$$P(\text{Tambunan}) = \frac{3}{13} = 0.231 \text{ (ada 3 halaman } web \text{ memiliki kata "Tambunan")}$$

Langkah 2 :

Cari *probability* antar kata kunci utama, yaitu $P(\text{Gayus} \cap \text{Tambunan})$.

$$P(\text{Gayus Tambunan}) = \frac{2}{13} = 0.154 \text{ (ada 2 halaman } web \text{ memiliki kata "Gayus Tambunan").}$$

Langkah 3 :

Cari nilai relevansi antar kata kunci

$$P(\text{Gayus} \mid \text{Gayus Tambunan}) = \frac{P(\text{Gayus} \cap \text{Tambunan})}{P(\text{Gayus})} = \frac{0.154}{0.538} = 0.286$$

$$P(\text{Tambunan} \mid \text{Gayus Tambunan}) = \frac{P(\text{Gayus} \cap \text{Tambunan})}{P(\text{Tambunan})} = \frac{0.154}{0.231} = 0.667$$

Dari hasil perhitungan nilai kata kunci diatas, dapat diketahui nilai-nilai *extended keyword* Gayus dan Tambunan terhadap kata kunci utama Gayus Tambunan menghasilkan data atau nilai sebagai berikut:

$$\text{Gayus Tambunan} = (1.0/\text{Gayus Tambunan}, 0.286/\text{Gayus}, 0.667/\text{Tambunan})$$

Kata kunci "Gayus Tambunan" yang merupakan kata kunci utama dinyatakan sebagai *fuzzy set* maka memiliki nilai 1.0 dikarenakan nilai relevansinya terhadap kata kunci itu sendiri sebagai kata kunci utama. Sedangkan kata kunci "Gayus" dan kata kunci "Tambunan" dinyatakan sebagai *extended*

keyword yang dihasilkan dari pemecahan kata kunci utama “Gayus Tambunan” memiliki nilai 0.286 untuk kata “Gayus” dan 0.667 untuk kata “Tambunan”.

4.1.2.2.3 Proses Perhitungan Halaman *Web* yang Memiliki Kata Kunci dan *Extended Keyword*

Setelah kata kunci “Gayus Tambunan” dimasukkan, maka akan diperoleh nilai probabilitas dari masing-masing kata kunci dan *extended keyword*. Setelah itu dilakukan perhitungan halaman *web* yang memiliki kata kunci dan *extended keyword*.

Perhitungan nilai halaman *web* yang mengandung kata kunci dicari menggunakan rumus TF-IDF (*Term Frequency – Inverse Document Frequency*). Perhitungan ini menggunakan rumus persamaan 2.6 sampai dengan persamaan 2.7 pada bab 2.

Perhitungannya manual nilai halaman *web* menggunakan kata kunci Gayus Tambunan sebagai berikut :

Perhitungan *web* terhadap kata kunci utama yaitu kalimat “Gayus Tambunan”

Tabel 4. 2 Jumlah kemunculan kalimat Gayus Tambunan terhadap halaman *web*

Halaman <i>Web</i>	Jumlah
1	2
2	1
3	0
4	0
5	0
6	0
7	0

Dari data di atas dapat dicari nilai t_{ij} pada halaman *web* pertama yaitu,

$$tf_{ij} = 0.5 + 0.5 \times \frac{2}{2}$$

$$tf_{ij} = 1$$

$$W_{ij} = 1 \times \frac{13 \times 2}{3}$$

$$W_{ij} = 8.667$$

Nilai relevansi kata kunci utama “Gayus Tambunan” terhadap halaman *web* pertama yaitu $W_{ij} = 8.667$. Untuk hasil perhitungan kata kunci utama “Gayus Tambunan” terhadap halaman *web* lainnya dapat dilihat pada tabel dibawah ini.

Tabel 4. 3 Hasil Perhitungan halaman *web* terhadap kalimat Gayus Tambunan

Halaman <i>Web</i>	Jumlah	Nilai
1	2	8.667
2	1	3.250
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0

Perhitungan *web* terhadap *extended keyword* pertama yaitu kata “Gayus”

Tabel 4. 4 Jumlah Kemunculan kata Gayus terhadap halaman *web*

Halaman <i>Web</i>	Jumlah
1	12
2	14
3	9
4	2
5	2
6	2
7	2

Dari data di atas dapat dicari nilai t_{ij} pada halaman *web* pertama yaitu,

$$t_{ij} = 0.5 + 0.5 \times \frac{12}{14}$$

$$t_{ij} = 0.929$$

$$W_{ij} = 0.214 \times \frac{13 \times 12}{43}$$

$$W_{ij} = 3.370$$

Nilai relevansi *extended keyword* kata “Gayus” terhadap halaman *web* pertama yaitu $W_{ij} = 3.370$. untuk hasil perhitungan *extended keyword* kata “Gayus” terhadap halaman *web* lainnya dapat dilihat pada tabel di bawah ini.

Tabel 4. 5 Hasil Perhitungan halaman *web* terhadap kata Gayus

Halaman <i>Web</i>	Jumlah	Nilai
1	12	3.370
2	14	4.233
3	9	2.234
4	2	0.345
5	2	0.345
6	2	0.345
7	2	0.345

Perhitungan *web* terhadap *extended keyword* kedua yaitu kata “Tambunan”

Tabel 4. 6 Jumlah Kemunculan kata Tambunan terhadap halaman *web*

Halaman <i>Web</i>	Jumlah
1	3
2	1
3	1
4	0
5	0
6	0
7	0

Dari data di atas dapat dicari nilai t_{ij} pada halaman *web* pertama yaitu,

$$tf_{ij} = 0.5 + 0.5 \times \frac{3}{3}$$

$$tf_{ij} = 1$$

$$W_{ij} = 1 \times \frac{13 \times 3}{5}$$

$$W_{ij} = 7.8$$

Nilai relevansi *extended keyword* kata “Tambunan” terhadap halaman *web* pertama yaitu $W_{ij} = 7.8$. untuk hasil perhitungan *extended keyword* kata “Tambunan” terhadap halaman *web* lainnya dapat dilihat pada tabel di bawah ini.

Tabel 4. 7 Hasil Perhitungan halaman *web* terhadap kata Tambunan

Halaman <i>Web</i>	Jumlah	Nilai
1	3	7.800
2	1	1.734
3	1	1.734
4	0	0
5	0	0
6	0	0
7	9	0

Maka perhitungan akhir untuk menghasilkan nilai relevansi kata kunci terhadap halaman *web* secara keseluruhan adalah sebagai berikut :

Perhitungan halaman *web* terhadap k_1 sebagai kata kunci utama = Gayus Tambunan, k_2 sebagai *extended keyword* pertama = Gayus, dan k_3 sebagai *extended keyword* kedua = Tambunan

adalah dengan mengalikan hasil nilai frekwensi kemunculan kata dalam halaman *web* dengan nilai relevansi tiap kata kunci, dan menggabungkan semua hasil perkalian itu.

Tabel 4.8 Perhitungan akhir untuk menghasilkan nilai relevansi kata kunci terhadap halaman *web* secara keseluruhan

Halaman <i>Web</i>	K_1	K_2	K_3	Nilai Relevansi
1	8.667	3.370	7.800	14.834
2	3.250	4.233	1.734	5.618
3	0	2.234	1.734	1.796
4	0	0.345	0	0.099
5	0	0.345	0	0.099
6	0	0.345	0	0.099
7	0	0.345	0	0.099

4.1.2.2.4 Proses Pengurutan Halaman *Web*

Pengurutan halaman *web* merupakan langkah terakhir dari proses pencarian. Pengurutan halaman *web* dilakukan dengan mengalikan hasil dari perhitungan nilai relevansi kata kunci dengan hasil dari perhitungan halaman *web*

yang memiliki kata kunci utama dan *extended keyword*. Halaman *web* akan diurutkan berdasarkan hasil dari yang paling besar hingga yang paling kecil.

Dari perhitungan yang dilakukan pada sub bab sebelumnya, maka diperoleh hasil urutan hasil pencarian halaman *web* untuk kata kunci “Gayus Tambunan” pada sistem ini sebagai berikut :

Tabel 4. 9 Pengurutan hasil pencarian halaman *web* untuk “Gayus Tambunan”

Halaman <i>Web</i>	K ₁	K ₂	K ₃	Nilai Relevansi
1	8.667	3.370	7.800	14.834
2	3.250	4.233	1.734	5.618
3	0	2.234	1.734	1.796
4	0	0.345	0	0.099
5	0	0.345	0	0.099
6	0	0.345	0	0.099
7	0	0.345	0	0.099

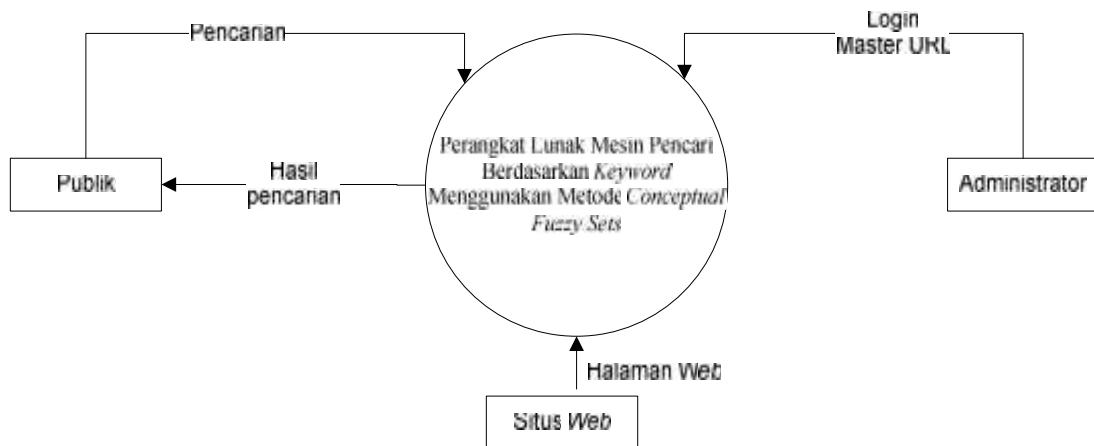
Data pada tabel di atas menjelaskan bahwa, halaman *web* 1 atau halaman *web* pertama memiliki nilai relevansi = 14.834. Halaman *web* ini memiliki nilai relevansi tertinggi dan menempati ranking pertama dalam pencarian menggunakan kata kunci Gayus Tambunan. Halaman *web* ini ditampilkan pada *list* pertama dalam halaman *web* pencarian pada perangkat lunak ini. Sedangkan halaman *web* 2 sampai dengan seterusnya akan berurut dibawah halaman *web* 1 sesuai dengan nilai relevansi dari terbesar hingga terkecil.

4.1.2.3 Deskripsi Fungsional

Analisa sistem dalam membangun Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets* menggunakan *Diagram Context*, *Data Flow Diagram* (DFD), *Entity Relational Diagram* (ERD) dan Bagan Alir (*FlowChart*).

4.1.2.3.1 Diagram Context

Diagram Context yang digunakan untuk mendeskripsikan proses aliran data Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets* dapat dilihat pada gambar 4.1 .



Gambar 4. 1 *Context Diagram*

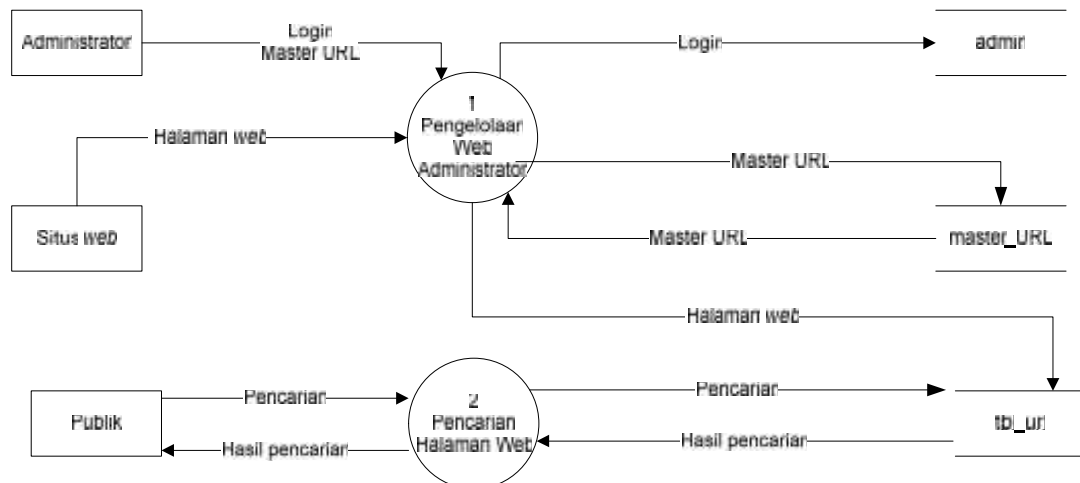
Entitas luar yang berinteraksi dengan sistem adalah Publik, Administrator dan Situs *Web*. Publik merupakan pengguna luar yang hanya dapat melakukan pencarian informasi di dalam aplikasi ini, Administrator adalah pengguna yang memiliki hak akses penuh terhadap aplikasi, baik dalam halaman *web* mesin pencari maupun dalam halaman *web crawler*, sedangkan situs *web* adalah entitas luar yang berfungsi sebagai pusat pembangun *database* serta sebagai media utama bagi publik dalam mendapatkan informasi yang lebih lengkap.

4.1.2.3.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika.

- a. DFD Level 1 Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets*.

Bentuk dari DFD Level 1 Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets* adalah sebagai berikut:



Gambar 4. 2 DFD Level 1 Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets*

Keterangan proses dan aliran data pada DFD level 1 Perangkat Lunak Mesin Pencari Berdasarkan *Keyword* Menggunakan Metode *Conceptual Fuzzy Sets*.

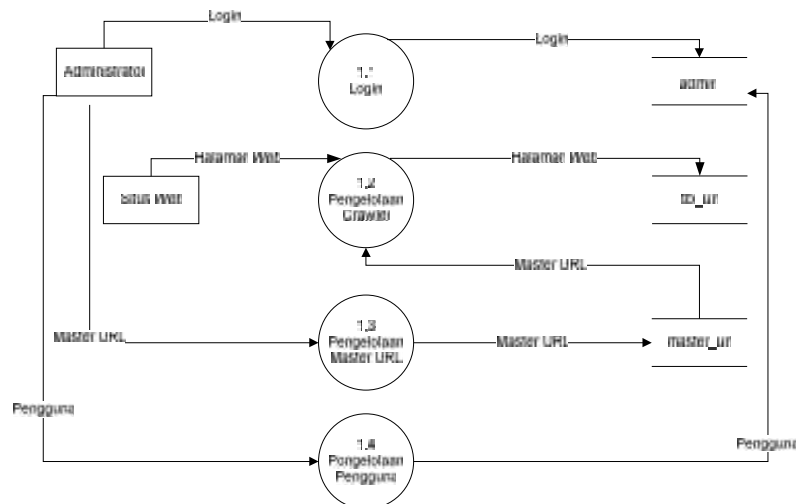
Tabel 4.10 Proses DFD Level 1 Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets*

Nama	Deskripsi
Pengelolaan Web Administrator	Proses yang melakukan pengelolaan <i>web</i> administator dan memiliki fungsi sebagai <i>crawler</i> yang bekerja membangun <i>database</i> untuk membantu proses pencarian halaman <i>web</i> , mengelola data master.
Pencarian Halaman Web	Proses yang bisa digunakan oleh Publik (pengguna internet) untuk melakukan pencarian halaman <i>web</i> .

Tabel 4.11 Aliran Data DFD Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets*

Nama	Deskripsi
Login	Data yang berisi ID <i>admin</i> dan <i>password</i> admin
Master URL	Data yang meliputi URL domain dari situs <i>web</i>
Halaman <i>Web</i>	Data yang berisi halaman web yang dipanggil oleh <i>crawler</i>
Pencarian	Data yang berisi kata kunci yang dimasukkan oleh publik.
Hasil Pencarian	Data yang berisi halaman web yang memiliki kata kunci yang dicari oleh publik

b. DFD Level 2 Proses Pengelolaan *Web* Administrator.



Gambar 4. 3 DFD 2 Level Proses Pengelolaan *Web* Administrator

Keterangan proses dan aliran data pada DFD level 2 proses Pengelolaan *Web* Administrator

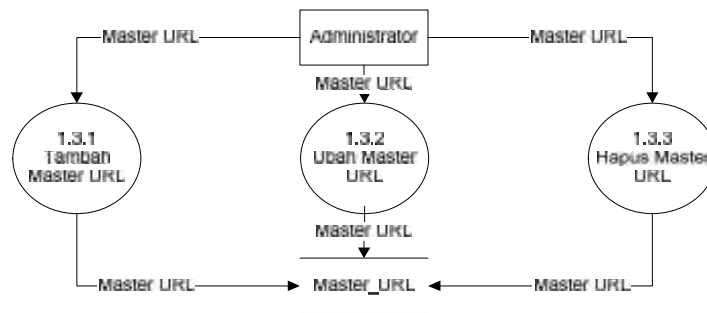
Tabel 4.12 Proses DFD Level 2 Proses Pengelolaan *Web* Administrator

Nama	Deskripsi
Login	Berisi proses login yang dilakukan oleh Administrator
Pengelolaan Crawler	Berisi proses pengambilan halaman <i>web</i> dari Situs <i>web</i>
Pengelolaan Master URL	Berisi proses pengelolaan data master URL
Pengelolaan Pengguna	Berisi proses pengelolaan data Administrator

Tabel 4.13 Aliran Data DFD 2 Proses Pengelolaan *Web Administrator*

Nama	Deskripsi
Login	Data yang berisi ID Admin dan <i>password</i>
Halaman <i>Web</i>	Data yang berisi halaman web yang dipanggil oleh <i>crawler</i>
Master URL	Data yang meliputi URL halaman <i>web</i> utama dari situs <i>web</i>
Pengguna	Data yang berisi data login administrator

c. DFD Level 3 Proses Master URL



Gambar 4. 4 DFD Level 3 Proses Master URL

Keterangan proses dan aliran data pada DFD level 3 Proses Master URL

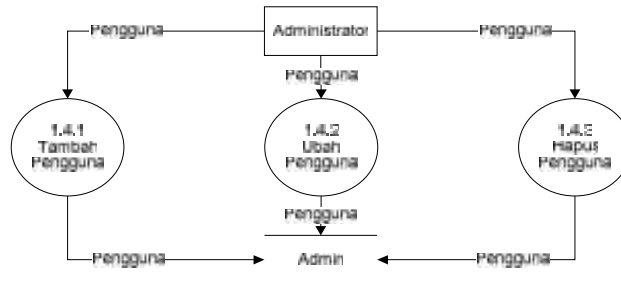
Tabel 4.14 Proses DFD Level 2 Proses Master URL

Nama	Deskripsi
Tambah Master URL	Berisi proses tambah data master URL
Ubah Master URL	Berisi proses ubah data master URL
Hapus Master URL	Berisi proses hapus data master URL

Tabel 4.15 Aliran Data DFD 2 Proses Master URL

Nama	Deskripsi
Master URL	Berisi data Master URL yang terdiri dari Kode master, URL.

d. DFD Level 3 Proses Pengelolaan Pengguna



Gambar 4.5 DFD Level 3 Proses Pengguna

Keterangan proses dan aliran data pada DFD level 3 Proses Pengguna

Tabel 4.16 Proses DFD Level 2 Proses Pengguna

Nama	Deskripsi
Tambah Pengguna	Berisi proses tambah data Pengguna
Ubah Pengguna	Berisi proses ubah data Pengguna
Hapus Pengguna	Berisi proses hapus data Pengguna

Tabel 4.17 Aliran Data DFD 2 Proses Pengguna

Nama	Deskripsi
Pengguna	Berisi data Id admin dan <i>Password</i> yang ditambahkan

4.1.2.3.3 Kamus Data

Pada kamus data ini, semua jenis data yang terlibat dalam proses yang terjadi, didefinisikan dan dikumpulkan dalam bentuk penyajian seperti berikut :

Tabel 4.18 Kamus Data admin

Nama	: admin
Deskripsi	: Nama, <i>Password</i>
Bentuk Data	: Tabel atau File
Sumber/Tujuan	: Berasal dari administrator / agar dapat masuk dan menggunakan sistem dengan aman
Periode	: Setiap dilakukannya proses login
Volume	: Tergantung Input dari administrator

Tabel 4.19 Kamus Data Master *URL*

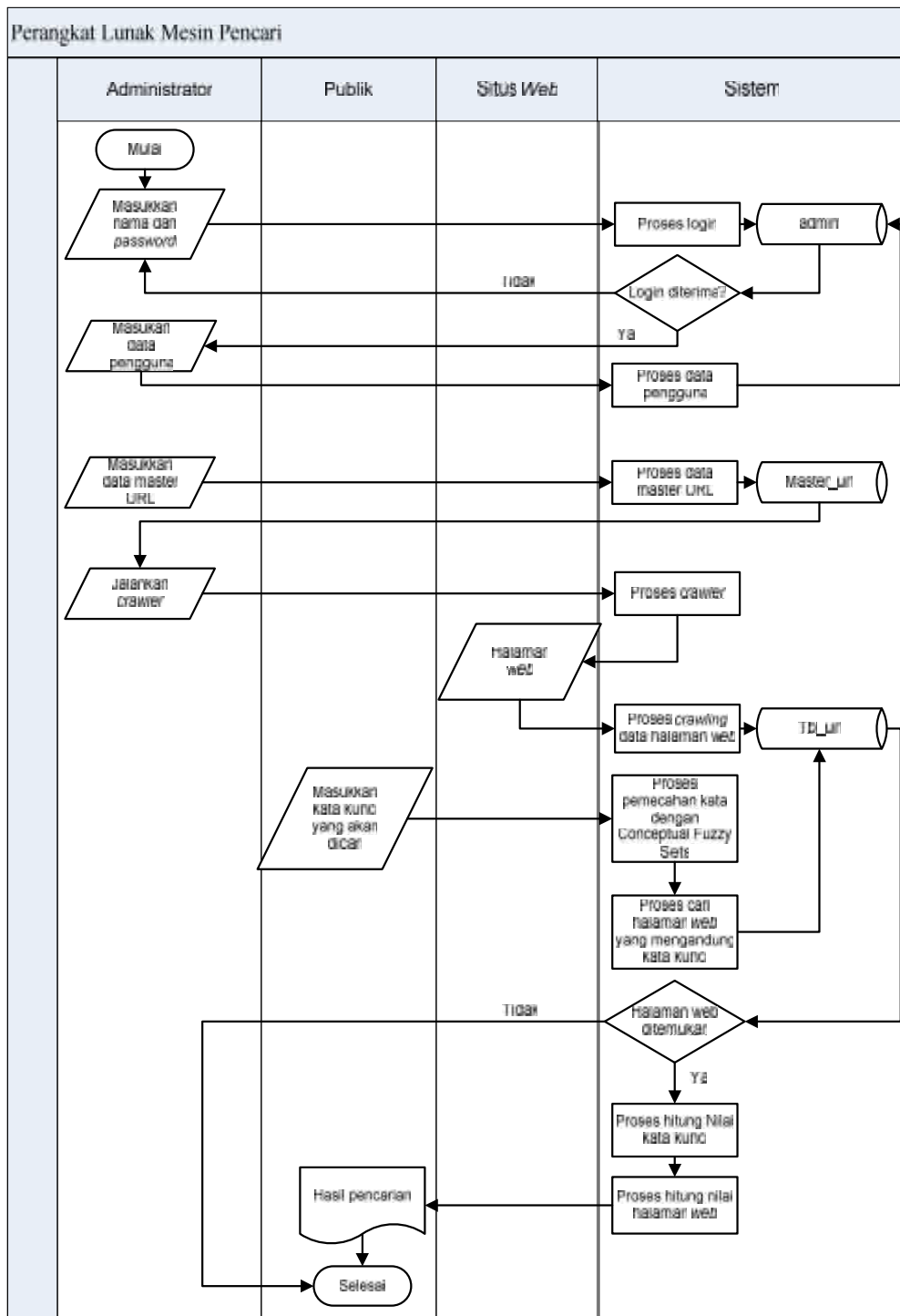
Nama	: master_url
Deskripsi	: Kode master, <i>URL</i>
Bentuk Data	: Tabel atau <i>File</i>
Sumber/Tujuan	: Berasal dari administrator / sebagai alamat situs utama yang akan dikunjungi oleh <i>crawler</i> dalam mengambil halaman-halaman <i>web</i> yang bertujuan untuk data dalam fasilitas pencarian.
Periode	: Setiap dilakukannya proses penambahan, perubahan dan penghapusan data, serta ketika proses menjalankan <i>crawler</i>
Volume	: Tergantung Input dari Administrator

Tabel 4.20 Kamus Data *Crawler*

Nama	: tbl_url
Deskripsi	: Hasil dari proses <i>crawler</i>
Bentuk Data	: Tabel atau <i>File</i>
Sumber/Tujuan	: Berasal dari proses crawling halaman web oleh <i>crawler</i> / sebagai data pencarian halaman <i>web</i>
Periode	: Setiap dilakukan updating <i>crawler</i> dan pencarian informasi oleh publik
Volume	: Tergantung sistem <i>crawling</i> yang dilakukan oleh proses <i>crawler</i>

4.1.2.3.4 Bagan Alir (*Flowchart*)

Bagan ini menjelaskan urutan-urutan dari prosedur yang ada didalam sistem dan menunjukkan apa yang dikerjakan sistem dan pengguna.



Gambar 4.6 Bagan Alir Perangkat Lunak Mesin Pencarian Menggunakan Metode *Conceptual Fuzzy Sets*

4.2 Perancangan

4.2.1 Perancangan Tabel

4.2.1.1 Tabel *admin*

Tabel 4.21 Tabel *admin*

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>
Nama	<i>Varchar</i>	25
<i>Password</i>	<i>Varchar</i>	25

Keterangan :

1. nama : Nama administrator untuk *login* ke aplikasi
2. Password : *Password* yang digunakan administrator untuk *login*

4.2.1.2 Tabel *master_url*

Tabel 4.22 Tabel *master_url*

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>
kode_master	<i>Integer</i>	5
URL	<i>Text</i>	

Keterangan :

1. kode_master : Kode master dari tabel *master_url*
2. URL : Alamat halaman *web*

4.2.1.3 Tabel *tbl_ur*

Tabel 4.23 Tabel *tbl_url*

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>
Id	<i>Integer</i>	5
URL	<i>Varchar</i>	500
Html	<i>Longtext</i>	
Status	<i>Enum ("0" dan "1")</i>	
Body	<i>longtext</i>	

Keterangan :

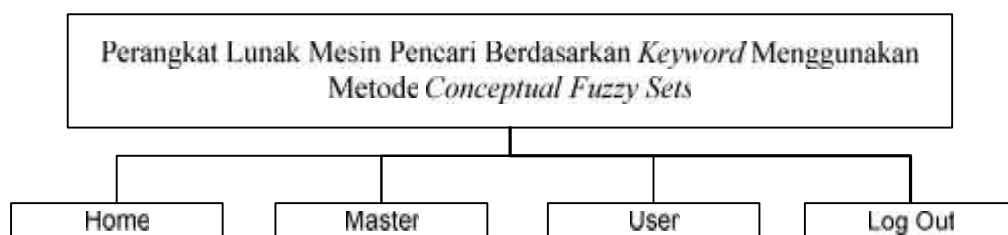
1. id : id dari tabel *crawler*
2. URL : Alamat *page*
3. html : isi informasi dalam halaman *web*

4. status : informasi suatu halaman *web* yang telah atau belum di-
crawling
5. body : isi dari dari halaman *web*

4.2.2 Perancangan Antar Muka

Rancangan antar muka sistem adalah sarana pengembangan sistem yang digunakan untuk membuat komunikasi yang baik, dan konsisten antara sistem dengan pemakainya. Penekanan rancangan antar muka meliputi tampilan yang baik, mudah dipahami, tombol-tombol yang *familiar* serta *user friendly*. Tujuan perancangan adalah untuk membuat panduan pada tahap implementasi mengenai rancangan dari sistem yang akan dibuat, supaya implementasi dapat dilakukan secara modular tetapi tetap konsisten.

Untuk memudahkan pemakaian sistem diperlukan susunan daftar pilihan atau menu sehingga pengguna yang belum terbiasa dengan sistem juga dapat menggunakan sistem ini. Melalui perancangan antar muka inilah sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dirancang. Pengguna akan dihadapkan pada berbagai alternatif menu yang ada. Dalam menentukan pilihannya, pengguna dapat menggunakan tombol tertentu dan setiap pilihan akan menghasilkan respon atau jawaban tertentu. Berikut digambarkan struktur menu dari sistem yang dirancang.



Gambar 4.7 Struktur Menu

4.2.2.1 Rancangan Antar Muka Untuk Administrator

4.2.2.1.1 Rancangan *Login*

Form *login* digunakan untuk masuk ke dalam program aplikasi. Dengan demikian hanya pengguna yang terdaftar sebagai administrator saja yang dapat mengakses aplikasi utama.

The wireframe shows a rectangular layout. At the top is a wide box labeled 'Banner'. Below it is a smaller box labeled 'Gambar'. Underneath is the text 'Silahkan Lakukan Login Untuk Mengakses Program Crawler'. This is followed by two rows of input fields. The first row is labeled 'ID Admin' with a 'Gambar' label to its left. The second row is labeled 'Password' with a 'Gambar' label to its left. At the bottom right are two buttons labeled 'Masuk' and 'Keluar'.

Gambar 4.8 Rancangan *Login*

Tabel 4.24 Deskripsi Perancangan Antarmuka *Home*

Nama objek	Jenis	Keterangan
Banner	Banner	Tampilan judul
Gambar	<i>Image</i>	Gambar-gambar seputar berita
ID Admin	<i>Text</i>	Memasukkan ID Administrator
Password	<i>Password</i>	Memasukkan <i>password</i>
Masuk	<i>Button</i>	Masuk ke aplikasi
Keluar	<i>Button</i>	Membatalkan <i>login</i> dan keluar menuju halaman <i>web</i> pencarian

Rancangan antarmuka lainnya dapat dilihat pada lampiran A.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Implementasi dan pengujian merupakan tahap yang dilakukan setelah tahap analisa dan perancangan selesai dikerjakan.

5.1 Implementasi

Implementasi merupakan kelanjutan dari tahap perancangan sistem yang telah didesain. Implementasi juga merupakan tahap pembangunan sistem menggunakan perangkat keras dan perangkat lunak yang telah ditetapkan.

Tujuan implementasi antara lain.

1. Menyelesaikan desain sistem yang ada dalam dokumen perancangan yang telah disetujui.
2. Menguji dan mendokumentasikan program-program atau prosedur-prosedur dari dokumen perancangan sistem yang telah disetujui.
3. Memastikan bahwa pemakai dapat mengoperasikan sistem yakni dengan mempersiapkan secara manual pemakai serta melatih pemakai.
4. Mempertimbangkan bahwa sistem memenuhi permintaan pemakai yakni dengan menguji secara keseluruhan.
5. Memastikan bahwa konversi ke sistem baru berjalan dengan benar yakni dengan membuat rancangan, mengontrol dan melakukan instalasi sistem secara benar.

5.1.1 Lingkungan Implementasi

Lingkungan implementasi sistem ada 2 (dua) yaitu: lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1. 1 Perangkat Keras

Perangkat keras yang digunakan dalam implementasi Perangkat Lunak Mesin Pencari Berdasarkan *Keyword* Menggunakan Metode *Conceptual Fuzzy Sets* ini menggunakan komputer dengan spesifikasi :

1. *Processor* : Intel Core 2 Duo 1.73 GHz
2. *Memory* : 2048 MB
3. *Harddisk* : 120 GB

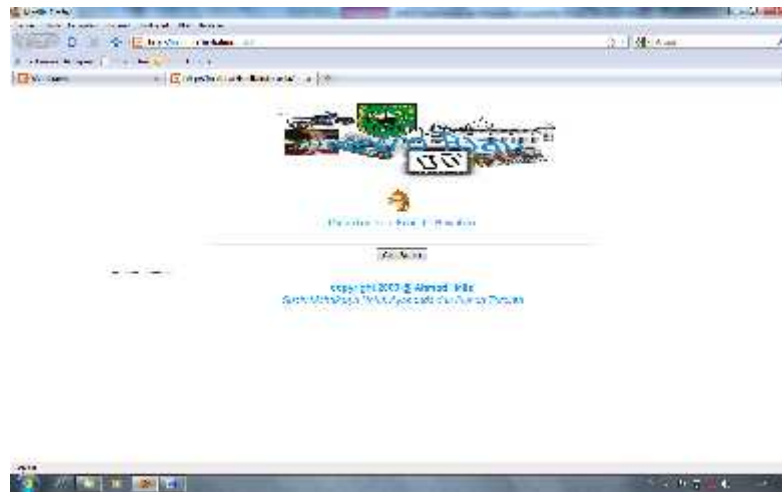
5.1.1. 2 Perangkat lunak

Perangkat lunak yang digunakan dalam implelementasi ini menggunakan:

1. Sistem Operasi : *Windows XP*
2. *Database Server* : MySQL 5.0.51b
3. *Web Server* : *Apache 2.2*
4. Pemrograman : PHP 5.2.6
5. *Web browser* : *Firefox Mozilla 4.0*

5.1. 2 Hasil Implementasi

Hasil implementasi menampilkan halaman *web* mesin pencarian. Halaman *web* mesin pencarian ini memberikan fasilitas kepada Publik untuk melakukan pencarian informasi dengan memasukkan kata kunci (*keyword*) pada *textbox* yang tersedia. Berikut adalah menu publik (menu Halaman *Web* Mesin Pencari) :



Gambar 5.1 Halaman Web Mesin Pencari

Hasil implementasi menu lainnya dapat dilihat pada lampiran B.

5.2 Pengujian Sistem

Setelah tahap implementasi dilakukan maka dilanjutkan dengan pengujian dari implementasi yang telah dibuat. Tahap pengujian diperlukan agar dapat diketahui hasil dari program implementasi sistem sesuai dengan analisa sistem menggunakan metode *Conceptual Fuzzy Sets*. Sistem ini diuji dengan menggunakan metode pengujian *black box*. Metode *black box* yaitu metode pengujian yang menguji suatu sistem tanpa harus mengetahui proses internal yang berada pada sistem tersebut. Berikut ini akan dijelaskan tahap-tahap pengujian terhadap aplikasi yang dibuat.

5.2.1 Identifikasi dan Rencana Pengujian

Tabel 5. 1 Identifikasi dan Rencana Pengujian

No Uji	Kelas Uji	Butir Uji	Tingkat Pengujian	Jadwal
1	Pengujian proses perhitungan probabilitas	Menghitung nilai kata kunci terhadap kata kunci lainnya	Pengujian Unit	25/01/2011

Tabel 5. 2 Identifikasi dan Rencana Pengujian (Lanjutan)

No Uji	Kelas Uji	Butir Uji	Tingkat Pengujian	Jadwal
2	Pengujian proses perhitungan frekuensi kata kunci terhadap halaman <i>web</i>	Menghitung nilai halaman <i>web</i> terhadap kata kunci	Pengujian Unit	25/01/2011
3	Pengujian proses perhitungan metode <i>Conceptual Fuzzy Sets</i> secara keseluruhan	Menghitung nilai <i>Conceptual Fuzzy Sets</i> akhir	Pengujian Unit	25/01/2011
4	Pengujian keseluruhan menu aplikasi	Pengujian apakah seluruh proses dalam menu-menu pada aplikasi berjalan sesuai dengan analisa yang telah dibuat.	Pengujian Unit	25/01/2011

Pengujian selanjutnya dapat dilihat pada lampiran C.

BAB VI

PENUTUP

Bab ini merupakan bab penutup yang berisi kesimpulan dari tujuan awal dan saran yang diperlukan untuk mengembangkan Perangkat Lunak Mesin Pencari ini selanjutnya.

6.1 Kesimpulan

Dari penjelasan bab-bab sebelumnya dapat diambil kesimpulan bahwa :

1. Aplikasi mesin pencarian dapat digunakan oleh pengguna sebagai mana mestinya. Bagi administrator, dapat menjalankan *web crawler* yang bertugas mengunjungi situs-situs *web* dan menyimpannya ke dalam basis data, sedangkan bagi publik, dapat melakukan pencarian dengan memasukkan kata kunci yang diinginkan.
2. *Web Crawler* yang digunakan pada aplikasi mesin pencarian ini berhasil bekerja mengunjungi situs-situs *web* dan mengumpulkanya ke dalam basis data.
3. Aplikasi mesin pencarian dengan menggunakan metode *conceptual fuzzy sets*, berhasil dirancang dan dibangun dengan menghasilkan hasil yang diinginkan. Metode ini menghasilkan urutan ranking ranking halaman *web*, sesuai dengan halaman *web* yang memiliki kedekatan makna paling tinggi terhadap kata kunci, hingga ranking paling rendah.

6.2 Saran

Dalam pengembangan perangkat lunak mesin pencarian dengan menggunakan metode *conceptual fuzzy sets* selanjutnya, penulis menyarankan agar :

1. Perangkat lunak mesin pencarian ini tidak hanya melakukan pencarian informasi terhadap halaman *web* saja, tetapi juga dapat melakukan pencarian terhadap gambar dan *file-file* tertentu.
2. *Web Crawler* yang digunakan pada perangkat lunak mesin pencarian, dapat menelusuri seluruh situs yang ada di dunia tanpa harus di-*input*-kan oleh administrator data alamat utamanya.

DAFTAR PUSTAKA

- Adith, “*Cara Kerja Spider Dalam Search Engine (Information Technology Articiel)*” [Online] Available <http://www.nucreativa.com/2009/01/29/google-page-rank/>, diakses tanggal 11 Januari 2011
- Christie, Jessica, “*Cara Cerdas Optimalkan Search Engine*”, Duta Wacana, Yogyakarta, 2009.
- Fanny, “*Pengantar Teknologi Informasi (Artikel Search Engine)*” [Online] Available <http://one.indoskripsi.com/judul-skripsi-tugas-makalah/pengantar-teknologi-informasi/artikel-search-engine>, diakses tanggal 11 Januari 2011
- Firdaus, “*PHP dan MySQL dengan Dreamweaver*”, Maxicom, Palembang, 2007.
- Hakim, Lukmanul, “*Membongkar Trik Rahasia Para Master PHP*”, Lokomedia, Yogyakarta, 2008.
- <http://bakauheni.com/en/category-table/286-mengetahui-dan-melihat-cara-search-engine-bekerja.html>
- http://devoav1997.webnode.com/news/teknik-pencarian-efektif-dengan-google/newsbcm_30419/180/
- <http://getete.blogspot.com/2007/12/search-engine.html>
- http://id.wikipedia.org/wiki/Mesin_pencari
- Muliantara, Agus, “*Penerapan Regular Expression Dalam Melindungi Alamat Email dari Spam Robot Pada Konten Wordpress* ” [Online] Available <http://ejournal.unud.ac.id/abstrak/3.%20penerapan%20regular%20expression.pdf> , diakses tanggal 11 Januari 2011
- Perkasa, Citra Dwi, “*Topical Wb Crawlers*” [Online] Available <http://www.hansmichael.com/default.asp?cat=exta201114391>, diakses tanggal 11 Januari 2011
- Setiawan, Ricky, “*Perancangan dan Pembuatan Search Engine dengan mengambil informasi dari World Wide Web dengan Metode Conceptual Fuzzy Sets*”, [Online] Available http://dewey.petra.ac.id/jiunkpe_dg_7415.html, diakses tanggal 11 Januari 2011

- Widyantoro, Dwi H, “*Survey Arah Penelitian, Pengembangan dan Penerapan Penjelajah Situs Web*” [Online] Available <http://www.batan.go.id/sjk/eII2006/Page05/P051.pdf>, diakses tanggal 11 Januari 2011
- Yan, Jun, dan Teman-teman, “*Using Fuzzy Logic*”, Practice Hall, New York, 1993.
- Zadeh, Lotfi, “*The Fuzzy System Handbook*”, AP Propfessional, New York, 1993.
- Zadeh, Lotfi, “*Fuzzy Logic and the Internet*”, Springer, New York, 2004.
- Zaiane , Osmar R, “*From Resource Discovery to Knowledge Discovery on the Internet (Artikel Mesin Pencari)*” [Online] Available http://118.98.171.131/webs/Artikel/Teknologi%20Informasi/Internet-wiki/mesin_pencari/, diakses tanggal 11 Januari 2011