ANALISIS DAN PERANCANGAN SEARCH ENGINE DOKUMEN PAPER BERBASIS WEB

NASKAH PUBLIKASI

diajukan oleh

Aldrik Saddermi

10.11.4055

kepada

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
AMIKOM YOGYAKARTA
YOGYAKARTA
2014

NASKAH PUBLIKASI

ANALISIS DAN PERANCANGAN SEARCH ENGINE DOKUMEN PAPER BERBASIS WEB

disusun oleh

Aldrik Saddermi

10.11.4055

Dosen Pembimbing

Krisnawati, S.Si, MT
NIK. 190302038

Tanggal, 24 November 2014

Ketua Jurusan
Teknik Informatika

Sudarmawan, MT
NIK. 190302035

ANALISIS DAN PERANCANGAN SEARCH ENGINE DOKUMEN PAPER BERBASIS WEB

Aldrik Saddermi[1)], Krisnawati[2)],

[1)] *Teknik Informatika STMIK AMIKOM Yogyakarta*
Jl Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta Indonesia 55283
Email : aldrigx@gmail.com[1)], krisna@amikom.ac.id[2)]

Abstract - ***The writer designed a search engine that focuses on finding paper documents using Focused Crawler search algorithm as a basis for the data content. Only the data on the papers and articles in Indonesia that will be the source of the search result. In addition, the search result is being filtered first so that the accuracy can be increased.***

Testing result show that the focused crawler is designed only to take certain information, thus reducing bandwidth and size data that enters into the database. While the path-ascending crawling take as much information as can be from destination website using indexing depth level 2.

Keywords *– Search Engine, Crawler Algorithm, PHP & MySQL*

1. Pendahuluan

- Latar Belakang Masalah

Apabila kita melakukan pencarian di mesin pencari khususnya di Google untuk file format tertentu seperti format *file* PDF, maka user harus mengetikan "kata kunci filetype:format file" dan tentunya hal ini kurang efektif karena tidak semua orang mengetahui cara ini

Sehingga pembuatan mesin pencari atau *search engine* ini, penulis menggunakan *Focused Crawler* sebagai dasar untuk memperoleh dokumen paper yang dibutuhkan. Sehingga pengumpulan data bisa dilakukan secara mudah dan hasil pencarian akan lebih spesifik karena web *crawler* bekerja secara otomatis memasuki dan menyimpan semua informasi yang terkandung di dalamnya.. Oleh karena itu penulis mengambil judul skripsi "Analisis dan Perancangan *Search Engine* Dokumen Paper Berbasis Web".

- Rumusan Masalah

Bagaimana merancang sebuah aplikasi yang dapat mengoptimalkan pencarian dokumen dengan pengelolaan *query* tanpa melalui mesin pencari Google dan Bing ?

- Batasan Masalah
  - Aplikasi dapat mencari dokumen yang relevan berdasarkan *query* user.
  - Pemrograman dilakukan menggunakan PHP *programming language*
  - *Search Engine* berbasis *Focused Crawler*
  - Tidak membahas keamanan web
  - Tidak membahas penilaian atau ranking dalam *search engine*
- Tujuan Penelitian

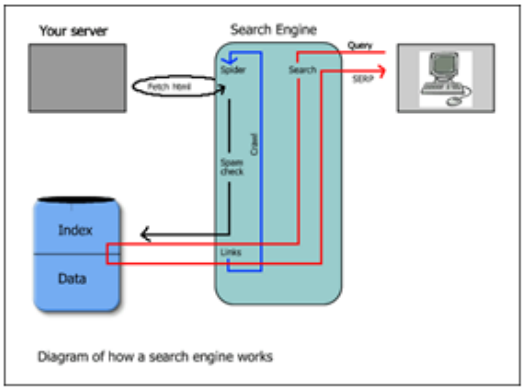
Merancang suatu aplikasi yang berbasis *web* yang dapat mengoptimalkan hasil pencarian dokumen yang berbasis pada algoritma *focused crawler*.

- Metode Penelitian

Metode yang digunakan adalah kepustakaan, studi literatur dan diskusi.

2. Landasan Teori

Fungsi umum *search engine* adalah mempermudah manusia memproleh informasi, bahkan untuk memenuhi beberapa kebutuhannya tanpa terbatas waktu dan tempat. [1]

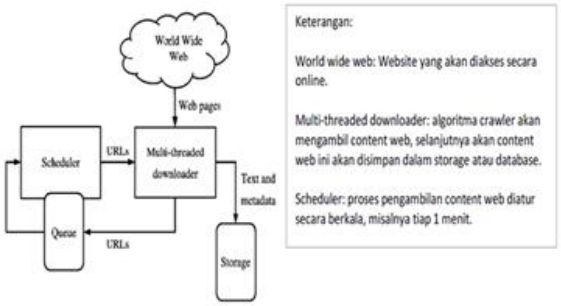


Gambar 1. Cara Kerja *Search Engine*

Search engine seperti yang banyak kita kenal, tidaklah benar-benar melakukan pencarian ke seluruh *World Wide Web* (www) secara langsung. Masing-masing pencarian dilakukan dalam database yang menyimpan *text* dari masing-masing halaman yang ada. *Text* dari halamanan demi halaman disimpan ke dalam server database. Ketika melakukan pencarian web dengan menggunakan *search engine*, yang dilakukan adalah pencarian salinan halaman yang disimpan pada database *search engine* itu sendiri yang berisi salinan halaman tersebut pada saat terakhir mereka kunjungi. Ketika meng-*click link* yang disediakan oleh halaman hasil pencarian yang dilakukan oleh *search engine*, sebenarnya alamat tersebut diberikan dari server *search engine* melalui versi terbaru yang ada didalam database *search engine* tersebut.

Database yang ada pada *search engine* dipilih dan dijaring oleh program robot yang disebut dengan *spider*. Meskipun *search engine* menjaring halaman yang akan diambil dan disimpan kedalam databasenya, dalam kenyataannya bisa dikatakan bahwa *search engine* mengambil dari suatu tempat. Kemudian untuk menemukan halaman potensial lainnya, *search engine* mengacu pada *link-link* yang terdapat pada halaman-halaman yang telah disimpan didalam database tadi.

Untuk *website* yang benar-benar baru dan belum ada satu pun *website* lain yang membuat *link* ke *website* baru tersebut maka spider pun tidak akan mengenalnya. Cara untuk membuat agar *website* baru tersebut bisa terdaftar pada *search engine*, khususnya untuk yang belum dapat *link* dari lain adalah dengan cara memberi tahu langsung *search engine* tersebut bahwa ada *website* baru yang pengerjaannya oleh manusia. [1]



Gambar 2. Arsitektur *Web Crawler*

Mengingat bahwa *bandwith* untuk melakukan *crawl* terbatas, maka penting untuk menjelajahi web tidak hanya dengan cara yang terukur tetapi efisien. Sebuah *crawler* harus hati-hati memilih langkah untuk halaman berikutnya. Perilaku web *crawler* dibuat berdasarkan kebijakan tertentu [2] yaitu :

- *Focused Crawling*

Page/halaman yang penting dan *page-page* yang berhubungan atau memiliki kesamaan akan menjadi fokus untuk dijelajahi web *crawler*. *Focused Crawler* didesain hanya untuk mengambil informasi tertentu saja, sehingga mengurangi trafik jaringan dan besar data yang dimasukkan kedalam database. Sebuah *Focused Crawler* atau *topical* web *crawler* idealnya hanya men-*download* halaman web hanya yang relevan dengan topik tertentu dan menghindari men-*download* halaman lain. Oleh karenanya *focused crawler* dapat memprediksi probabilitas bahwa *link* ke halaman tertentu adalah relevan sebelum benar-benar men-*download* halaman.

- *Restricting Followed Link*

Web *crawler* akan membatasi HTML *page* saja yang akan dikunjungi dan menghindari MMEtyle (*internet media type*) yang lain.

- *URL Normalization*

Menghindari *crawling page* yang sama dari satu sumber lebih dari satu kali.

- *Path-ascending Crawler*

Sebuah teknik *crawling* yang mengambil dan menjelahi sebuah halaman web sampai ke *path* direktori yang masih berhubungan. Algoritma ini memungkinkan *crawler* untuk menelusuri *page* yang tidak relevan, untuk mendapatkan *page* yang relevan. Prinsip kerja dari algoritma ini adalah untuk tetap menelusuri *page* sampai dengan batas kedalaman maksimum level yang ditentukan dari *page* yang tidak relevan

3. Analisis dan Perancangan

- Analisis Sistem
  - Identifikasi masalah

Penulis membahas mengenai analisis dan implementasi *focused crawler* dengan menggunakan *Content Similarity* dan *Link Structure Analysis*. *Content Similarity* merupakan salah satu metoda untuk menghitung kesamaan/kemiripan *content* sehingga cocok untuk diterapkan pada sistem berbasis *focused crawler* yang tujuan-nya mengumpulkan *page* yang relevan sesuai dengan topik yang ditentukan. *Link Structure* merupakan suatu metoda untuk menghitung *link*s *ranking* (*link score*) yaitu *link* – *link* yang berprioritas tinggi yang akan menunjukkan ke *page* yang relevan. Selain itu, pada sistem ini juga diimplementasikan metoda untuk menulusuri *link* – *link* yang terdapat pada *page* yang tidak relevan (*irrelevant*), sehingga dapat mencegah kemungkinan terbuangnya *link page* relevan yang berada di bawah *page* yang tidak relevan (*irrelevant*). Dengan kombinasi seperti itu diharapkan dapat mengumpulkan *page* relevan yang banyak sesuai dengan topik yang akan dicari. [3]

  - Analisa Kelemahan Sistem

Metode PIECES dapat digunakan untuk menganalisis masalah dan kelemahan pada sistem. PIECES sendiri meliputi Kinerja (*Performance*), Informasi (*Information*), Ekonomi (*Economic*), Kontrol (*Control*), Efisiensi (*Efficiency*), dan Pelayanan (*Service*).

  - Analisa Kebutuhan Sistem

Untuk mempermudah analisis tersebut dalam menentukan kebutuhan secara lengkap, maka analisis kebutuhan dibagi dalam dua jenis. Kebutuhan Fungsional dan Kebutuhan Non Fungsional.

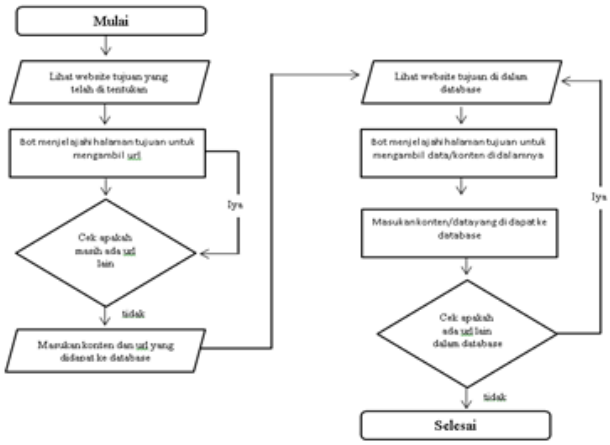
- Perancangan Sistem

Perancangan aplikasi dibagi kedalam dua tahap. Tahap pertama, aplikasi mengambil konten dari *website* tujuan dan memasukkannya kedalam *database*. Tahap kedua merupakan tahap dimana *user* menggunakan aplikasi yang sudah berisi data yang ada.

Perancangan proses atau pemodelan proses digunakan untuk menggambarkan bagaimana *search engine* dokumen paper ini berjalan. Perancangan proses pada aplikasi *website* ini, menggunakan pemodelan fisik (*phisycal model*) dengan membuat flowchart dan

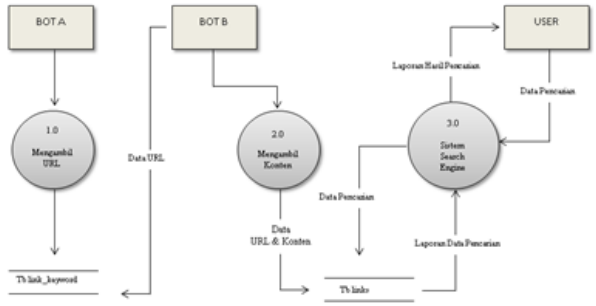
pemodelan lojik (*logical model*) dengan membuat DFD (Data Flow Diagram).

- Perancangan Flowchart



Gambar 3. Flowchart Aplikasi

- Perancangan Data Flow Diagram (DFD)

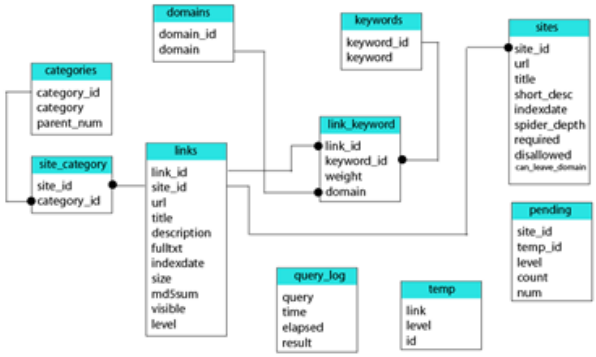


Gambar 4. DFD

- Perancangan ERD (Entity Relation Diagram)

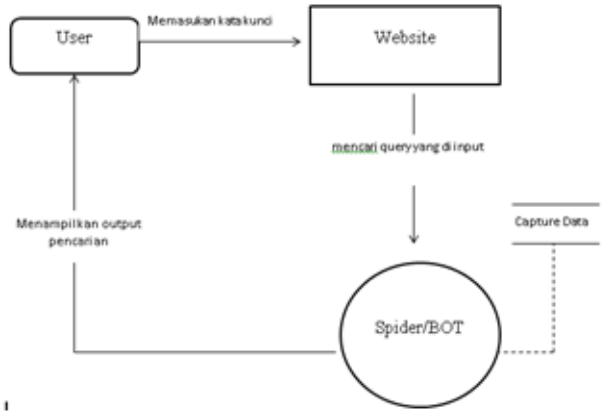
Gambar 5. Gambar ERD

- Rancangan Relasi Tabel



Gambar 6. Relasi Tabel

- Perancangan *Focused Crawler Non* Database



Gambar 7. Cara Kerja Bot Non Database

- Perancangan Interface

  - Halaman Utama
  - Halaman Hasil Pencarian (*Result Page*)
  - Halaman Login Admin
  - Halaman Dashboard Admin

- Perancangan Struktur Aplikasi

Gambar 8. Gambar Struktur Aplikasi untuk admin

4. Implementasi dan Pembahasan

Kegiatan implementasi dilakukan dengan rencana yang telah dibuat, dengan tahapan sebagai berikut :

- Pembuatan Database
- Pembuatan Interface (*Form*)
- Pembuatan Koneksi Database dengan *interface*
- Pembuatan *Coding* Program (*Source Code*)

Gambar 9. Antar Muka Main *Page*

Gambar 10. Antar Muka Hasil Penelusuran

Tahapan setelah dibuat program adalah pengujian.

- Pengujian Program

Kesalahan dari program yang mungkin terjadi dapat diklarifikasikan dalam tiga bentuk kesalahan, yaitu :

- Kesalahan Logika (*Logic Errors*)
- Kesalahan Bahasa (*Language Erros*)
- Kesalahan Dalam Proses (*Run Time Errors*)

- Pengujian Sistem

Ada dua metode untuk melakukan uji coba yaitu:

- Uji Coba *White Box*

  Adalah cara pengujian dengan melihat kedalam modul untuk meneliti kode-kode program yang ada dan mengalisa apakah ada kesalahan atau tidak.

- Uji Coba *Black Box*

  Merupakan tahap yang berfokus pada pernyataan fungsional perangkat lunak. Pengujian black box berusaha menemukan kesalahan dalam beberapa hal, diantaranya fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses database eksternal, kesalahan kinerja, installasi dan kesalahan transmisi.

Pada uji coba *black box testing*, terdapat kesalahan pada fungsi navigasi halaman pencarian yang tidak bisa berjalan semestinya. Fungsi tersebut menampilkan *error* karena kesalahan penulisan kode program. Kesalahan penulisan kode program terletak pada bagian parameter yaitu ‘addmark’, sehingga fungsi navigasi pada halaman hasil penelusuran tidak dapat dijalankan. Penulisan parameter yang benar seharusnya ‘addmarks’.

Tabel 1. Black Box Testing

No	Kelas Uji	Butir Uji	Hasil
1	Halaman Admin	Proses login dan validasi	OK
2	Halaman Utama	*Form Search Box with Database*	OK
		Form Search Box Focused Crawler non Database	OK
3	Halaman Pencarian	Menampilkan hasil pencarian	OK
		Navigasi Halaman Pencarian	*ERROR*
		Proses pencarian berdasarkan kategori	OK
4	Modul Situs	Proses tambah, *edit*, hapus situs	OK
5	Modul Kategori	Proses tambah, *edit*, hapus kategori	OK
6	Modul Index	Proses pengambilan URL dan Konten	OK
7	Modul Clean Table	Menghapus semua kata kunci yang tidak berhubungan dengan *link*	OK
		Menghapus semua kata kunci yang tidak berhubungan dengan situs	OK

- Pengujian penggunaan algoritma *Crawler*

Setelah dilakukan 2 percobaan terhadap 2 URL yang berbeda, bisa terlihat bahwa *Focused Crawler* dan *Path-Ascending Crawler* mempunyai kecepatan waktu yang sama dalam penelusuran . Namun jika dibandingkan dengan *Path-Ascending Crawler* yang mengambil semua URL yang ada dalam halaman tersebut, *Focused Crawler* terlihat lebih efektif dimana dia hanya mengambil URL yang diinginkan.

Tabel 2. Pengujian Algoritma *Crawler*

Percobaan	URL	Focused *Crawler* (waktu(m)/jumlah URL)	Path-Ascending *crawler* (waktu(m)/jumlah URL)
1	belajarpsikologi.com	7/250	7/306
2	ilmumanajemen.wordpress.com	8/127	8/233

- Pengujian Perbandingan Sistem

Setelah dilakukan sepuluh percobaan pada tabel dibawah, bisa terlihat jika *website* dapat melakukan tugasnya dalam mencari dan menampilkan hasil pencarian yang dilakukan dengan baik. Mesin pencari yang menggunakan *database* terlihat lebih cepat dalam menampilkan data dan hasil jumlah pencarian lebih sedikit. Sebaliknya mesin pencari yang tanpa menggunakan *database* mempunyai kecepatan yang lama dalam menampilkan data karena bergantung pada koneksi internet yang digunakan oleh *user*. Sedangkan hasil pencariannya jauh lebih banyak karena menggunakan data yang bersumber dari google books secara langsung. Pada percobaan nomer 9 kata kunci kenapa tidak bisa ditemukan oleh mesin pencari yang menggunakan *database*, karena kata kunci ini termasuk kata kunci yang diabaikan oleh sistem sebagai *ignored words.*

Tabel 3. Pengujian Perbandingan Kecepatan Sistem

Percobaan	Kata Kunci	*Search Engine* with database (s/jumlah pencarian)	*Search Engine* non database (s/jumlah pencarian)
1	akuntansi	0,01 /69	0,89 /11.500
2	psikologi	0,02 /252	0,52 /11.400
3	contoh skripsi	0,01/251	0,23/10
4	metode pengumpulan data	0,1/23	0,21/90
5	ilmu	0.09 /319	0,27/132.000
6	kumpulan makalah	0,09/76	0,13/211
7	pengertian akuntansi	0,04/69	0,25/239
8	pajak	0,07/68	0,65/239
9	kenapa	0,01/tidak ditemukan	0,16/7.600
10	sistem informasi	0,18/99	0,45/3.790

- Manual Installasi

Pada manual instalasi akan dijelaskan bagaimana cara menginstal atau menjalankan aplikasi *search engine* ini pada komputer atau laptop.

- Manual Program

Manual program yaitu penjelasan mengenai tatacara dalam menggunakan program yang dibuat untuk memudahkan pengguna program agar tidak terjadi kesalahan dalam pengoperasiannya.

5. Kesimpulan

Hasil pengujian *crawler* menunjukan bahwa *focused crawling* lebih efektif untuk digunakan jika dibandingkan dengan *path-ascending crawling*. Dilihat dari jumlah URL yang didapat lebih terarah dan valid. Dan Sistem *focused crawler* ini bisa diimplementasikan pada kasus lain, misal untuk mengumpulkan *content* multimedia seperti gambar, video dan audio.

Daftar Pustaka

[1] Febrian, Jack. *Google & Yahoo! Secret!*. Bandung : Informatika. 2007.

[2] Anonim [online] "*A Web Crawler*". https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Web_*crawler*.html

[3] Christopher Olston and Marc Nojark, *Web Crawling (Foundations and Trends in Information Retrieval*), Boston : Now Publisher Inc, 2010.

Biodata Penulis

Aldrik Saddermi, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2014.

Krisnawati, memperoleh gelar Sarjana Sains (S.Si), MIPA Ilmu Komputer UGM. Memperoleh gelar Magister Teknik (M.T) Program Pasca Sarjana Magister Sistem Komputer & Informatika Fakultas Teknik Elektro Universitas Gajah Mada Yogyakarta. Saat ini menjadi Dosen di STMIK AMIKOM Yogyakarta.