

APLIKASI SEARCH ENGINE DENGAN METODE DEPTH FIRST SEARCH (DFS)

Noni Juliasari¹⁾, Joseph Christian Sitompul²⁾

¹⁾²⁾ Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260, Indonesia
e-mail : noni.juliasari@budiluhur.ac.id¹⁾

Abstract

Along with the times and technology, it turns out that the demand for information is increased. The information required should also be obtained quickly and accurate. From a variety of technologies to obtain information, one of which is search engine. The role of technology search engine becomes increasingly important in information technology, especially in obtaining the up-to-date information. For example, sometimes we want to know what information is held by an area in the rest of the world, because maybe the area is an area of our prospects or a resource area. This study will discuss the development of simple search engine application by using depth first search method. It is hoped this this assessment will be useful and will be better in the future.

Keywords: Search Engine, DFS, URL

1. PENDAHULUAN

Saat ini, telah menjadi rahasia umum bahwa salah satu kunci keberhasilan dalam suatu kegiatan adalah informasi. Dimana informasi yang dibutuhkan tersebut juga bukan sekedaranya saja melainkan dibutuhkan juga kecepatan dan keakuratan dalam proses perolehannya, karena informasi yang sudah lawas atau tidak *up to date* tidak akan terlalu bermanfaat terutama untuk dasar pengambilan suatu keputusan dalam sebuah organisasi. Namun permasalahan yang timbul adalah adanya keterbatasan. kemampuan sumber daya manusia dalam proses perolehan informasi yang dibutuhkan tersebut. Sehingga dalam penelitian ini kami mencoba membangun sebuah aplikasi *search engine* sebagai salah satu sarana yang dapat memudahkan manusia dalam mencari informasi tersebut.

Pembangunan aplikasi ini tujuan utamanya adalah sebagai alat yang memudahkan pengaksesan informasi dalam sebuah portal atau situs tertentu, kemudian tujuan lainnya adalah juga untuk memberi kecepatan dalam pengaksesan terhadap informasi tersebut. Namun

aplikasi dibatasi hanya untuk melakukan pencarian *file* atau informasi ke dalam lingkungan internet dengan menggunakan metode *depth first search* (DFS).

2. TINJAUAN PUSTAKA

2.1 Search Engine

Search engine merupakan mesin pencari yang berupa sebuah *website* untuk mencari informasi yang tersimpan di dalam situs yang lain. Tiga tugas dasar sebuah *search engine* dalam cara kerjanya :

- Mencari di internet atau memilih bagian-bagian dari internet menurut kata-kata penting atau kunci
- Memberi indeks pada kata-kata yang dicari, dan dimana mereka menemukannya
- Mengijinkan pengguna untuk mencari kata-kata atau kombinasi kata yang ditemukan pada indeks.

2.2 Web Crawler

Sebuah *web crawler* (atau dikenal juga dengan nama lain *web spider* atau *web robot* atau *web scutter*) adalah sebuah program atau skrip otomatis yang

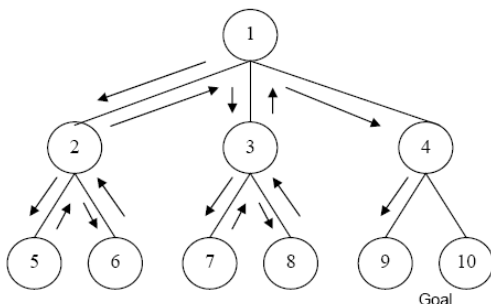
menjelajah ke area *world wide web* untuk membuat *copy* dari sebuah halaman web yang sudah dikunjungi untuk selanjutnya oleh *search engine* akan dilakukan *indexing* agar proses pencarian menjadi cepat. *Crawler* dapat juga digunakan untuk melakukan *maintenance* otomatis seperti memeriksa *link-link* atau validasi kode *html*.

2.3 Reguler Expression

Reguler Expression adalah notasi yang dapat digunakan untuk mengelola teks (*describe and parse text*). *Reguler Expression* mirip sebuah bahasa pemrograman kecil (*mini programming language*) yang sangat hebat, fleksibel dan efisien dalam memproses data-data teks. *Reguler Expression* sering digunakan kalau kita melakukan fungsi "*search and replace*", *searching world wide web*, atau mengolah data-data. *Reguler Expression* banyak diimplementasikan di berbagai bahasa seperti PHP, javascript, database mysql, lisp, perl, python dan masih banyak yang lainnya.

2.4 Depth First Search

DFS singkatan dari *Depth First Search* adalah algoritma pencarian pada sebuah pohon atau *tree*. Pencarian DFS ini adalah dengan menelusuri satu cabang sebuah pohon sampai ke bawah (menemukan solusi) sebelum melakukan *backtracking*. Ilustrasi pencarian dengan DFS seperti ditunjukkan gambar 2.1, pencarian dimulai dari akar (level 0) dan pencarian dilanjutkan dengan melacak *node* yang berada paling kiri.



Gambar 1 *Depth First Searching*

3. METODA

Metodologi yang digunakan pada penelitian ini adalah sebagai berikut:

a. Studi literatur

Sebelum memulai penelitian, dilakukan studi literatur, baik dari buku-buku, jurnal-jurnal, maupun artikel-artikel yang berhubungan dan menunjang penelitian ini.

b. Menyusun metode

Hasil dari studi literatur akan diimplementasikan menjadi suatu metode untuk proses pembangunan aplikasi.

c. Mengembangkan program aplikasi

Metode yang dihasilkan akan diaplikasikan pada sebuah program untuk menguji keefektifannya.

d. Menguji metode

Metode yang dihasilkan akan diuji dengan implementasi program aplikasi yang dikembangkan. Seandainya pada hasil pengujian ditemukan kekurangan-kekurangan metode yang dikembangkan, maka dilakukan perbaikan dan kembali ke langkah b (menyusun metode). Demikian seterusnya sampai metode yang dikembangkan dianggap cukup efektif. Secara garis besar, keefektifan metode diukur terhadap penggunaan atau pemanfaatan teknologi.

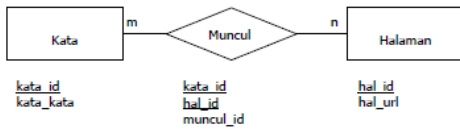
4. HASIL DAN DISKUSI

Aplikasi *search engine* yang dibangun melakukan penyimpanan terhadap indeks-indeks dari seluruh halaman web yang ada. *Database* indeks dari halaman web tersebut juga akan ter-*update* oleh bantuan *web crawler*.

Web crawler digunakan dengan maksud utama untuk menciptakan sebuah *copy* dari semua halaman-halaman web yang telah dikunjungi untuk pemrosesan lebih lanjut oleh *search engine* yang akan memberikan indeks halaman-halaman web yang telah di-*download* untuk mempercepat proses pencarian. *web crawler* adalah salah satu tipe bot atau *software agent*.

Secara umum, *web crawler* memulai pekerjaannya dengan sebuah daftar dari

url-url yang akan dikunjungi, yang disebut *seeds*. Ketika *web crawler* mengunjungi url-url tersebut, *web crawler* mengidentifikasi semua *hyperlinks* dalam halaman web tersebut dan menambahkannya pada daftar url yang akan dikunjungi.

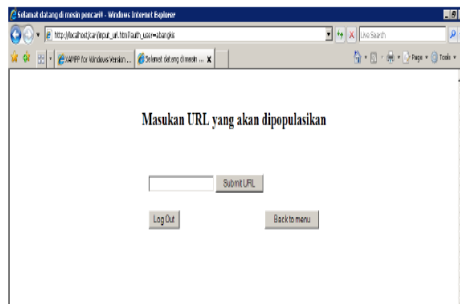


Gambar 2. Rancangan ER Diagram

Beberapa modul utama yang terdapat dalam aplikasi antara lain adalah :

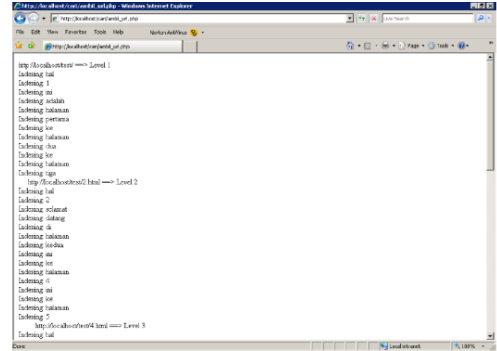
a. Modul input URL

Menu ini adalah awal dari program koleksi URL, dimana seorang *data entry* harus mengisi URL-URL yang akan diperiksa oleh sistem ini.



Gambar 3 Tampilan modul input URL

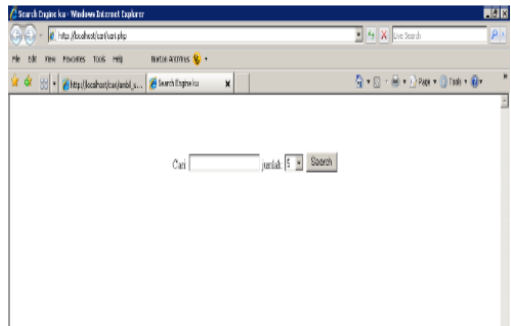
Bila URL yang diinput oleh *data entry* dapat dibuka, maka URL tersebut akan ditampilkan dan kata-kata yang *diparsing* akan disimpan dalam database cari dan ditampilkan. Begitu juga pada URL level berikutnya.



Gambar 4. Tampilan hasil proses input URL

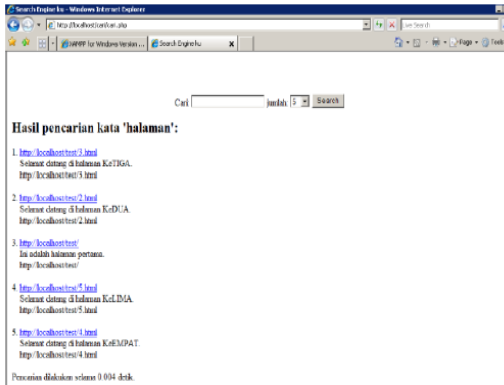
b. Modul pencarian

Menu cari adalah halaman awal dimana seorang *user* akan memasukkan kata kunci dari informasi yang ingin dicari dan menentukan berapa alamat URL yang memuat kata tersebut yang akan ditampilkan.



Gambar 5. Tampilan modul pencarian

Bila pada proses pencarian kata di database berhasil ditemukan, maka yang akan ditampilkan adalah alamat URL tempat kata tersebut berada, banyaknya kemunculan kata tersebut pada alamat URL tersebut dan waktu yang dibutuhkan untuk proses pencarian tersebut.



Gambar 6. Tampilan hasil proses pencarian

Adapun kekurangan dan kelebihan aplikasi *search engine* yang dikembangkan ini adalah sebagai berikut:

1) Kelebihan

- Program ini sangat mudah digunakan untuk mengumpulkan url, kata dan kemunculan sebuah kata.
- Program ini juga dapat mendeteksi apakah sebuah url dapat dibuka atau tidak.
- Kata yang sama tidak akan disimpan lebih dari sekali pada *database* sehingga *database* tidak akan *overload*. Melainkan hanya indeks kata saja yang disimpan dengan indeks muncul. Dengan demikian akan meringankan kinerja *server*.
- Pencarian kata yang terkandung dalam URL sangat mudah dilakukan.

2) Kekurangan

- Sehubungan adanya kemungkinan sebuah URL memiliki anak URL yang memiliki link yang panjang, maka pencarian kata yang dilakukan hanya dibatasi sampai URL level 3.
- Search engine* ini baru mampu mencari halaman web per 1 kata. Bila diisi lebih dari satu kata maka hasil pencarian tidak muncul.

5. KESIMPULAN

Dari hasil analisa terhadap aplikasi *search engine* dengan menggunakan metode *depth first search*, maka dapat ditarik beberapa kesimpulan, antara lain:

- Proses pemeriksaan URL dapat dilakukan dengan cepat dan diketahui aktif tidaknya suatu URL.
- Proses *indexing* yang dilakukan meringankan beban *database* dan mempercepat proses pencarian kata.
- pencarian kata dapat dilakukan dengan cepat dan tepat.

DAFTAR PUSTAKA

- Herrington, Jack D., PHP HACKS, Tips & Tools for Creating Dynamic Web Sites, O'Reilly, First Edition, Sebastopol, 2005.
- Haryanto, Steven, PHP Kumpulan Resep Pemrograman, Dian Rakyat, Jakarta, 2004.
- Siswoutomo, Wiwit, PHP Undercover, Mengungkap Rahasia Pemrograman PHP, PT.Elexmedia Komputindo, Jakarta, 2005.
- McCarty, Bill, PHP 4, A Beginner's Guide, Osborne, Jakarta, 2001.
- Syafii, M., Membangun Aplikasi berbasis PHP dan MySQL, Andi Offset, Yogyakarta, 2004.