Analisis Penerapan Algoritma DFS dan BFS pada Web Crawling/Spider

Abel Stanley / 13517068
Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132, Indonesia
13517068@std.stei.itb.ac.id

***Abstract*—Penyimpanan data telah sekian lama berpindah dari penyimpanan analog menjadi penyimpanan digital. Hampir seluruh data di dunia ini disimpan pada Internet. Informasi-informasi di internet menjadi sangat besar dan perlu sistem untuk menangani pengambilan dan pemilihan data, yaitu melalui Information Retrieval Systems (IR). Alat penelusuran yang populer di zaman sekarang seperti Google, Yahoo search, Bing, Baidu bergantung pada Web Crawlers untuk membangung Web Index. Web crawlers akan menjelajahi graf web, mengunduh laman-laman, dan mencari alamat-alamat web menuju ke laman baru yang ingin dijelajahi. Untuk melakukan hal tersebut, Web Crawlers dapat menerapkan algoritma *Breadth First Search dan Depth First Search*. Pada makalah ini akan dibandingkan performa dan hasil yang didapat melalui setiap algoritma tersebut.**

Keywords—Web Crawling, Information Retrieval Systems (IR), Breadth First Search, Depth First Search, Page Rank

I. PENDAHULUAN

Pada zaman sekarang, Web menjadi salah satu media komunikasi dan interaksi antara individual manusia yang paling cepat dan efektif. Aplikasi yang dibasiskan pada Web tentunya memiliki *demand* yang semakin tinggi. Salah satu aplikasi berbasis Web adalah Search Engines seperti Google, Yahoo Search, Baidu, dan Bing. Sayangnya, algoritma yang secara spesifik mendefinisikan cara kerja search engines dari perusahaan-perusahaan terdepan seperti contoh diatas masih terselubung menjadi rahasia perusahaan. Namun, secara fundamental, search engines tersebut menerapkan algoritma pencarian graf dasar seperti BFS dan DFS yang telah dimofidikasi dan dioptimisasi sesuai kebutuhanya.

Karena popularitas search engine telah berkembang secara eksponensialsejak beberapa tahun ini, pelayanan yang optimal menjadi aspek yang penting bagi penyedia search engine. Waktu yang dilakukan untuk melakukan search harus sangat singkat dan tentunya harus menghasilkan hasil laman yang akurat dan paling terbaru. Oleh karena itu, pemilihan algoritma yang digunakan menjadi tahap yang sangat krusial untuk memaksimumkan efektifitas dan efisiensi dari performa search engine.

II. DASAR TEORI

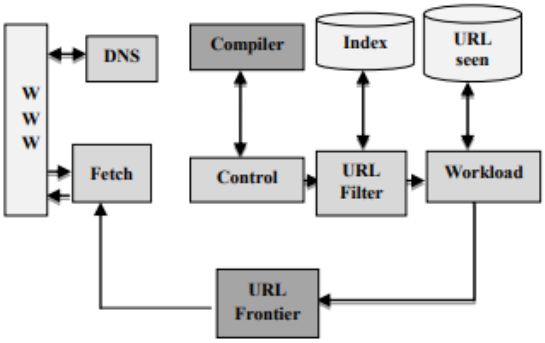
2.1 Web Crawling

Web Crawling adalah pendekatan yang mengubah data yang tidak terstruktur menjadi data yang terstruktur. Hal ini mengeksploitasi karakteristik dari struktur HTML, seperti metadata, anchor, dll., untuk mengumpulkan informasi. Dua pendekatan yang umum digunakan adalah DFS (Depth First Search) dan BFS (Breadth First Search). Web crawler tidak akan bisa mengunduh setiap laman dari hasil query search engine karena ada limitasi hardware, bandwidth, dan limitasi jaringan lainnya. Web Crawler juga harus menuruti aturan server yaitu dengan mematuhi kriteria dari robot.txt file dan tidak boleh overload web servers.

Pada umumnya, Web Crawlers memiliki 5 sektor utama yaitu:

1. Decision Maker Module
2. Fetching Module
3. Control Module
4. Filter Module
5. Workload Module

Decision Maker Module (DNS) akan mencari IP address pada nama-nama domain. Modul ini akan menentukan dimana laman yang diinginkan akan diakuisisi. Compiler yang diawasi oleh Control Unit akan meng-seed laman-laman dan mengirimkan dokumen-dokumen ke Filter Module yang menggunakan protokol HTTP untuk memulihkan laman-laman. Text Filter Module bekerja dengan mengekstraksi beberapa links dari laman-laman yang di-fetch. Setelah separasi, alamat-alamat web yang sesuai akan dikirim ke Workload Unit dan akan diletakkan pada list instruksi selanjutnya untuk melakukan *fetch unit*. Filter Unit memiliki dua sektor, yaitu link filtering dan indexing. Berikut grafik yang menerangkan fungsi utama dari Web Crawler.



Gambar 1. Arsitektur dari Web Crawler

Sumber: Tajbar-Parashkoohi, Saeideh & Ahmadi-Abkenari, Fatemeh

2.2. Page Rank

Page Rank adalah pengukuran popularitas dari setiap laman di internet. Page Rank adalah algoritma untuk menganalisis alamat-alamat web untuk menentukan derajat kepentingan dan kesesuaian dari sebuah laman website. Hasil pengukuran dari Page Rank bersifat statik dan digunakan sebagai ganti dari queries. Hal ini berarti Page Rank akan menghitung nilai "Kepentingan Global" dari setiap laman.

Approach yang umum digunakan untuk menghitung Global Worth adalah Link Based Approach. Hyperlink yang menunjuk ke laman suatu website akan menaikkan popularitas dari laman tersebut. PageRank dari suatu laman akan dihitung secara rekursif dan bergantung pada angka dan metrik penilaian PageRank dari setiap laman yang terhubung dengannya berdasarkan link. Jadi dari pengertian ini, sebuah laman yang terhubung dengan banyak laman yang memiliki Page Rank yang tinggi akan memiliki nilai Page Rank yang tinggi juga. Page Rank diberi nama dari Larry Page dan digunakan oleh Google Web Search Engine yang memberikan nilai-nilai weight yang bersifat numerik kepada laman-laman.

Misalkan sebuah laman bernama X memiliki laman sebanyak T1…Tn yang menunjuk padanya. Parameter d adalah damping factor. Nilai dari d bisa diberi nilai dari 0 hingga 1. Untuk kasus perusahaan Google, damping factor bernilai 0.85

Rumus yang digunakan untuk menghitung Page Rank:

PR(X) = (1-d) + d(PR(T1)/C(T1) + … + PR(Tn)/C(Tn))

Dimana:

1. PR(X) adalah PageRank dari Laman X
2. PR(Ti) adalah PageRank dari laman Ti yang menunjuk pada laman X
3. C(Ti) adalah banyaknya link keluar dari halaman Ti
4. d adalah damping factor yang bisa diatur dari nilai 0 hingga 1

Dari rumus diatas, bisa disimpulklan bahwa PageRank tidak memberikan ranking kepada laman-laman internet secara keutuhan, namun ia akan menghitung PageRank dari setiap laman yang terhubung secara individu. Urutan dari Crawling tidak akan mempengaruhi PageRank. PageRank dari X akan ditentukan secara rekursif. oleh PageRank-PageRank dari setiap laman yang menunjuk pada laman X.

III. STRATEGI PENCARIAN GRAF

Struktur sebuah website dapat diilustrasikan sebagai suatu graf besar dan kompleks yang terdiri dari simpul-simpul yang melambangkan laman-laman web dan beberapa koneksi dan links dari setiap laman sebagai busur. Strategi melakukan website crawling dapat dibagi menjadi 3 kategori umum yaitu uninformed/blind search, heuristic/informed search, dan local search.

Dalam strategi Uninformed Search, informasi yang nyata dan mendefinisikan masalah dan keadaan target dibandingkan dengan keadaan non-target. Pada strategi ini, tidak ada informasi sama sekali mengenai bagaimana dan jalur manakah yang harus diambil untuk mencapai target. Dengan demikian, metode blind ini akan mencari sampai keseluruhan dari search space untuk mendapatkan target. DFS(Depth First Search), BFS(Breadth First Search) dan Uninformed Cost Search (UCS) adalah contoh dari Algoritma Uninformed Search.

3.1. Depth First Search

Algoritma Depth First Search pertama kali diperkenalkan di tahun 1994 dan diaplikasikan di Web Crawler sebagai algoritma terbaik untuk bertahun-tahun. Web Crawler yang menganut algoritma ini menelusuri alamat web dengan menggunakan sebuah frontier sebagai antrian FIFO(First In First Out). Control Unit dari Web Crawler akan menentukan sebuah laman sebagai titik awal pada suatu laman untuk *fetching unit*. Proses pergerakan dari laman ke laman akan lanjut terus hingga suatu level kedalaman telah dicapai. Ketika sebuah simpul dan anaknya telah diekstensi ke jalur tersebut, dia akan dihilangkan dari memori. Maka dari itu, algoritma ini memiliki kompleksitas memory yang linear, yaitu space complexity: O(bm), b adalah faktor percabangan atau maksimum banyaknya busur yang menjurus keluar dari sebuah simpul dan m adalah kedalaman dari pohon. Pada kasus terbutuk, algoritma ini akan mengekstensi ke setiap simpul dari graf yang sedang ditelusuri dimana kompleksitas waktu menjadi $O(b^m)$.

Masalah dengan kompleksitas waktu yang tinggi dapat dijumpai jika suatu jalur penelusuran ada yang indefinite / terlalu panjang, alias menjebak logika pencarian, sehingga solusi terkadang tidak dapat ditemukan akibat pemilihan jalur awal yang salah. Meskipun algoritma ini sederhana, banyak

laman yang berkualitas rendah berdasarkan konten yang tidak sesuai dengan laman utama akan disimpan di repository.

```
{Fungsi DFS}
DFS(G, u)
   u.traversed = true
   for each v ∈ G.Adjacent[u]
      if (v.traversed == false){
         DFS(G,v)
      }

{Menjalankan fungsi DFS}
main() {
   For each u in G :
      u.traversed = false
    For each u in G :
     DFS(G, u)
}
```

Figure 1: Pesudocode Depth First Search

3.2. Breadth First Search

Pada strategi Breadth First Search (BFS), setelah mengspesifikasikan seed dari laman, control unit akan menentukan setiap simpul dengan breadth level yang sama dan memperkenalkanya ke *fetching unit*. Setelah crawler menjelajahi setiap laman yang dispesifikasikan pada level itu, control unit akan pergi ke breadth level selanjutnya dan menganalisis simpul-simpul di level tersebut. Strategi ini lebih mudah didesain dan diimplementasikan. Karena ada limitasi dari banyaknya link keluar, ukuran dari repository tidak akan meningkat secara drastis. Pada Depth First Search, Fetching Unit akan menganalisis setiap link web pada suatu laman hingga ke kedalaman yang ditentukan dan akan pergi ke laman selanjutnya untuk mendapatkan informasi lebih lanjut.

```
 {Ubah setiap node menjadi “untraversed”}

  q = new Queue();

  q.insert(SimpulAwal);

  while ( q ≠ empty ) do
  {
    x = q.pop(); {mengambil elemen pertama
queue}

    if ( !traversed(x)  )
    {
      traversed[x] = true;       {kunjungi simpul
x}

      for ( setiap busur(x,y)   )
        if ( !traversed[y] )
              q.insert(y);
    }
  }
```

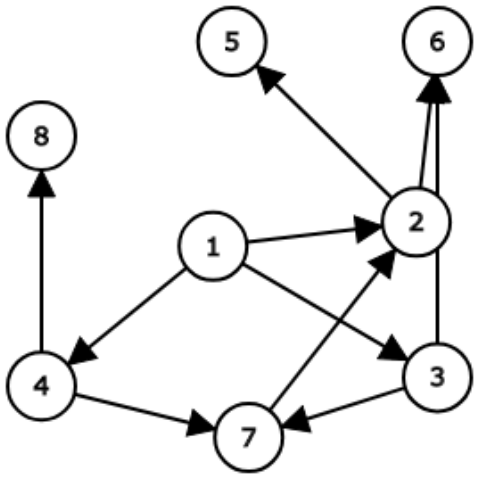
Figure 2: Pseudocode BFS Algorithm

IV. EKSPERIMEN

Jika web dianggap sebagai graf maka setiap laman website akan dianggap sebagai simpul-simpul dari graf dan setiap hyperlink sebagai busur penyambung dari sebuah graf. Misal ada sebuah website contoh yang seperti demikian:

Laman1:
Mempunyai link menunjuk ke: 2 3 4
Ditunjuk oleh: Tidak ada
Laman2:
Mempunyai link menunjuk ke: 5 6
Ditunjuk oleh: 1 7
Laman3:
Mempunyai link menunjuk ke: 6 7
Ditunjuk oleh: 1
Laman4:
Mempunyai link menunjuk ke: 7 8
Ditunjuk oleh: 1
Laman5:
Mempunyai link menunjuk ke: Tidak ada
Ditunjuk oleh: 2
Laman6:
Mempunyai link menunjuk ke: Tidak ada
Ditunjuk oleh: 2 3
Laman7:
Mempunyai link menunjuk ke: 2
Ditunjuk oleh: 3 4
Laman8:
Mempunyai link menunjuk ke: Tidak ada
Ditunjuk oleh: 4

Asumsi jika laman A merupakan laman “Seed”, maka graf yang terbentuk adalah sebagai berikut:



Gambar 2: Graf dari contoh struktur web

Maka jika diterapkan Algoritma PageRank akan menghasilkan hasil sebagai berikut:

Laman	Nilai PageRank
1	0.01874
2	0.02416
3	0.05739
4	0.04314
5	0.02405
6	0.03921
7	0.05337
8	0.02898

Tabel 1. Tabel Laman dan Page Rank berkorespondensi

Sebuah Web Crawler harus memiliki kondisi stop agar tidak menjelajahi dalam infinite loop. Dapat diterapkan algoritma BFS dan DFS untuk hal ini.

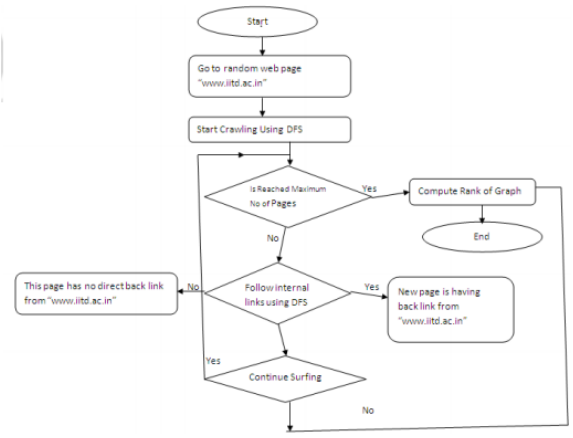
4.1. Depth First Search Approach:

Pada strategi Depth First Search, akan dilihat graf sebagai pohon. Pencarian akan dimulai pada halaman seed dan web crawler akan mencari mendalam terus-menerus hingga setiap halaman telah dijelajahi. Ketika sampai pada tahap seperti itu, DFS akan backtrack keatas dan lanjut menelusuri ke cabang laman yang lain. Hal ini berarti, sambil menelusuri laman-laman di web, akan dilihat link pertama dari setiap laman hingga dicapai laman yang buntu atau tidak ada link keluar lagi. Hanya jika hal tersebut sudah dilakukan baru pencarian akan berpindah ke link kedua dari laman awal dan laman yang terhubung.

Untuk DFS, urutan penelusuran adalah sebagai berikut:

1,4,8,7,2,5,3,6

Flowchart yang menunjukkan cara kerja perhitungan PageRank dengan DFS adalah sebagai berikut:



Flowchart 1. Sistem perhitungan rank dengan DFS

Sumber: Kumari, Poonam & Kakhani, Gaurav

Pada pencarian DFS ada satu kekurangan yang mencolok. Ketika kita menjelajahi internet, seharusnya urutan penelusuran laman internet oleh web crawler tidak mempengaruhi hasil laman-laman yang akan dijelajahi. Namun pada pencarian DFS, jika pada pencarian pertama misalnya tidak dapat menyelesaikan penjelajahan karena link laman ke laman lain selalu ada dan tidak pernah putus (endless chain) dan tidak ada kondisi untuk memberhentikan web crawler, maka penjelajahan web crawler akan gagal.

Oleh karena itu, strategi yang bisa dianut pada kasus penjelajahan DFS adalah dengan mendefinisikan sebuah threshold banyaknya laman internet yang boleh dijelajahi sebelum web crawling dilaksanakan, sehingga web crawler akan berhenti ketika telah menjelajahi laman sebanyak threshold yang diberikan.

Namun, kelemahan dari strategi ini adalah pada realita, kita tidak dapat memprediksi urutan dari web crawling. Hal ini akan mempengaruhi PageRank dari laman-laman sehingga menjadi tidak akurat.

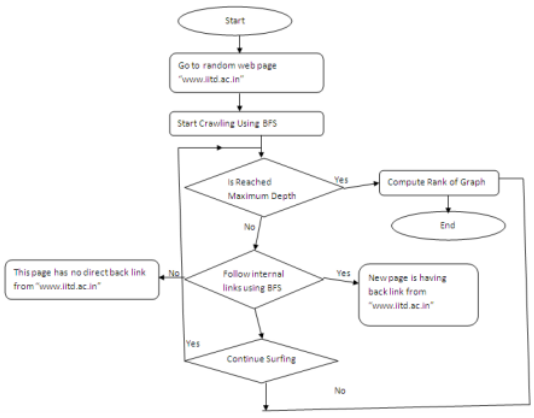
4.2. Breadth First Search Approach:

Pada penjelajahan Breadth First Search, kita tidak perlu mendefinisikan di awal kondisi stop dan besar threshold banyaknya laman yang bisa dijelajahi oleh web crawler. Pendekatan ini tidak menjelajahi graf laman internet dalam satu arah saja. BFS akan menjelajahi laman “seed” dan menyimpan setiap links yang memiliki kedalaman yang berbeda 1 level dari laman “seed” setelah itu baru menjelajah ke level selanjutnya.

Urutan penjelajahan pada graf website contoh yang diberikan adalah:

1, 2, 3, 4, 5, 6, 7, 8

Flowchart yang menunjukkan cara kerja sistem penentuan PageRank dengan strategi *Breadth First Search* adalah:



Flowchart 1. Sistem perhitungan rank dengan BFS

Sumber: Kumari, Poonam & Kakhani, Gaurav

Pada pendekatan ini, BFS akan dimulai dengan angka konstanta yang akan melambangkan kedalaman ke-berapa web crawler akan berjelajah. Algoritma ini bagus jika graf sudah distrukturkan dengan urutan yang memiliki kriteria bahwa simpul-simpul laman di kedalaman yang rendah adalah merupakan laman yang berkualitas tinggi. Dengan algoritma BFS, laman-laman berkualitas tinggi itu pasti akan dijelajahi dan tidak pernah terlewatkan. Setiap laman hingga kedalaman yang terspesifikasi pasti akan dijelajahi dan kedalaman tersebut dapat diatur sesuai kebutuhan.

V. HASIL EKSPERIMEN

Digunakan smallseotools untuk mengecek nilai Google PageRank dari beberapa website contoh yang akan dieksperimen. Website yang digunakan untuk mengecek nilai Google PageRank adalah:

https://smallseotools.com/google-pagerank-checker/

Hasil dari google PageRank akan dijadikan PageRank standar. Berikut nama website yang digunakan untuk eksperimen beserta PageRank Standarnya:

a. http://www.iitd.ac.in
   PageRank Standar: 8
b. http://nano.iitd.ac.in
   PageRank Standar : 7
c. http://www.fittiitd.org
   PageRank Standar : 7

Keterangan: PageRank standar menggunakan skala 1-10.

1. Depth First Search

Dijalankan program BFS web crawler dalam Bahasa python 3.6 pada titik-titik banyak laman yang telah dijelajahi secara incremental sebesar 500 laman. untuk di bandingkan hasil standar untuk mengecek urutan PageRank relatif dari setiap laman.

Laman	PageRank			
	500 laman	1500 laman	2500 laman	4500 laman
http://www.iitd.ac.in	1.6415 e-05	5.781 e-05	6.718 e-05	9.4965 e-05
http://nano.iitd.ac.in	Gagal	Gagal	Gagal	1.3328 e-05
http://www.fittiitd.org	1.6332 e-05	5.752 e-05	6.654 e-05	8.8387 e-05

Tabel 3. Hasil PageRank DFS

Kelemahan dari DFS adalah harus didefinisikan diawal banyaknya webpage yang boleh dijelajahi. Inkremen pada banyaknya laman diambil dengan acuan pada Depth level dari tabel BFS sehingga dapat dikomparasi dengan tepat.

Dapat dilihat dari hasil DFS bahwa ada beberapa website yang tidak dapat dijelajahi secara DFS karena terjebak pada penelusuran yang menuju depth yang indefinite.

2. Breadth First Search

Dijalankan program BFS web crawler dalam Bahasa python 3.6 pada depth maks yang berbeda-beda dari 1-5 untuk di bandingkan hasil standar untuk mengecek urutan PageRank relatif dari setiap laman.

Laman	PageRank				
	Depth 1	Depth 2	Depth 3	Depth 4	Depth 5
http://www.iitd.ac.in	0.000403	0.00281	0.00214	0.002803	0.002814
http://nano.iitd.ac.in	0.000174	0.00051	0.00024	0.000501	0.000501
http://www.fittiitd.org	0.000181	0.00052	0.00025	0.000514	0.000512

Tabel 4. Hasil PageRank BFS

Urutan relatif dari PageRank diantara ketiga laman ini sama seperti PageRank standar. Dengan demikian dinyatakan bahwa Breadth First Search menghasilkan hasil yang sesuai dan tepat.

VI. KESIMPULAN

Dilakukan eksperimen pengukuran PageRank pada beberapa contoh website yang dipilih secara random. Digunakan algoritma Google PageRank melalui smallseotools untuk mendapatkan nilai PageRank standar. Lalu juga didapatkan nilai PageRank dari program python 3.6 DFS dan BFS web crawler. Hasil menunjukkan bahwa Breadth First Search akan memberikan hasil yang lebih baik dan optimal berdasarkan nilai PageRank. Pada pendekatan BFS akan dijamin untuk selalu mendapatkan laman yang akurat dan sesuai keinginan namun pada kasus Depth First Search kita tidak bisa menjamin akurasi laman yang dijelajahi. Pada kasus terburuk laman yang “seed” bisa tidak dijelajahi. Jadi dapat disimpulkan bahwa BFS akan memberikan hasil yang lebih baik untuk web crawling.

VII. UCAPAN TERIMA KASIH

Penulis pertama-tama ingin memanjatkan puji dan syukur kepada Tuhan yang Maha Esa atas anugerah dan kesempatan yang diberikan untuk menulis makalah ini. Hanya karena berkat-Nya makalah ini dapat diselesaikan

tepat waktu. Selanjutnya, penulis juga ingin berterima kasih kepada seluruh dosen pengajar dari mata kuliah Strategi lagoritma atas kesempatan dan pengalaman beliau perbolehkan bagi penulis untuk alami dan pelajari. Dari kesempatan ini, penulis dapat belajar banyak mengenai format-format gambar dan teknik kompresi data. Penulis juga ingin berterima kasih kepada orang tua, keluarga, dan kerabat penulis karena tanpa mereka, penulis tidak akan menjadi diri penulis yang sekarang.

VIII.REFERENSI

[1] Castillo, Carlos & Nelli, Alberto & Panconesi, Alessandro. *Crawling the Web with Limited Memory*.

[2] Tajbar-Parashkoohi, Saeideh & Ahamdi-Abkenari, Fatemeh. 2014. *Page Quality Optimization in Crawler's Queue through Employing Graph Traversal Algorithms.* International Journal of Computer Applications

[3] California State University, Northridge. 2016. *SpyBite: A New Approach to Designing a Web Crawler.*

[4] Kumari, Poonam & Kakhani, Gaurav. 2013. *Comparative Analysis of Web PageRank Algorithm using DFS and BFS Crawling*. International Journal of Science and Research.

[5] http://www.cis.uni-muenchen.de/~yeong/Kurse/ss09/WebDataMining/kap8_rev.pdf, diakses pada 25 April 2019 pukul 19.42 WIB.

IX. PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 April 2019

Abel Stanley 13517068