

Penggunaan Algoritma Breadth First Search dalam Aplikasi Penduplikasian Situs

Philip Anderson¹, Wijoyo Simanjuntak²

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if13060@students.if.itb.ac.id¹, if14035@students.if.itb.ac.id²

Abstrak

Dalam mendapatkan informasi yang berharga dari internet, banyak cara yang bisa dilakukan oleh seorang user. Sebagian user membaca/melihat informasi yang tertuang dalam *situs-situs* internet tersebut lalu merangkum/menulisnya di dalam catatan, sebagian melakukan duplikasi terhadap sebagian *text/gambar/media* lainnya melalui cara lama (*copy-paste*), sebagian lagi *menyimpan* halaman *situs* tersebut dengan tujuan suatu saat dapat membukanya kembali tanpa harus mengakses internet lagi. Namun kendala yang terjadi adalah apabila *situs* yang akan kita *simpan* ada banyak dan punya banyak *link*, tentu hal ini membutuhkan ketelitian dan kesabaran yang tinggi, serta waktu dan ruang *penyimpanan* yang besar.

Pada penulisan makalah ini, penulis menyadari bahwa *penyimpanan* halaman-halaman *situs* yang cukup kompleks dan mempunyai banyak *link* tersebut dapat dibantu dengan *algoritma Breadth First Search* (Pencarian Melebar) yang telah diajarkan dalam kuliah IF2251 Strategi Algoritmik. Software yang dibuat dengan algoritma ini akan mengunjungi setiap halaman *situs* sampai kedalaman tertentu dan *menyimpannya* ke dalam media *penyimpanan*.

Kata kunci: *Algoritma Breadth First Search, situs, simpan, link*

1. Pendahuluan

Setiap orang pada umumnya pernah menggunakan internet. Apakah itu hanya untuk menghabiskan waktu luang, mencari informasi-informasi penting, membeli suatu barang melalui media internet, membaca pesan elektronik (e-mail), memesan tiket pesawat dan banyak hal baik yang bisa dilakukan dengan media internet.

Penggunaan internet tidak selamanya dilakukan untuk hal-hal yang bertujuan positif, justru sekarang telah banyak berkembang *situs-situs* yang memberikan nilai-nilai negatif kepada penggunaanya seperti *situs-situs* propaganda, pornografi, SARA, *situs-situs* yang memberitakan kabar bohong, bahkan *situs-situs* untuk melakukan kejahatan.

Dalam mendapatkan informasi dari *situs* tersebut, sebagian user *menyimpannya*. Memang untuk menyimpan satu atau dua halaman *situs* memang sangat mudah, cukup dengan memilih menu File – Save as. Dengan begitu, user dapat menentukan dan memilih halaman mana saja yang dianggap berguna dan yang pantas disimpan.

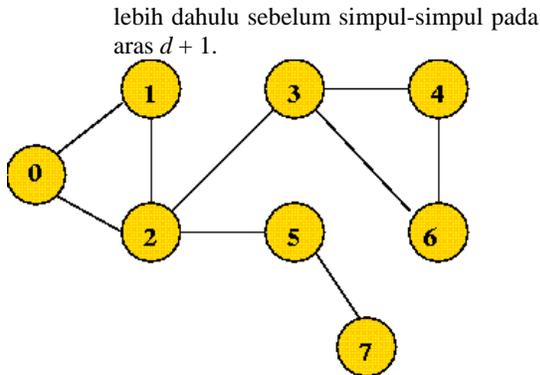
Tetapi berbeda halnya apabila *situs* yang dibuka itu memiliki banyak *link*, padahal informasi yang kita butuhkan tersebar di seluruh *link* yang terdapat

dalam *situs* tersebut, dengan kata lain, user ingin menyimpan seluruh *situs* ke dalam media penyimpanan yang dimiliki user. Disinilah algoritma Breadth First Search (Pencarian Mendalam) dipakai. Algoritma ini akan mengunjungi mulai dari *situs* induk (alamat *situs* yang pertama sekali kita masukkan), menyimpannya, kemudian membaca *link-link* yang terdapat dalam halaman induk tersebut dan mengunjunginya serta menyimpannya. Hal ini dilakukan sampai batasan tertentu yang merupakan input user.

2. Algoritma Breadth First Search

Mengutip Diktat Kuliah IF2251 Strategi Algoritmik yang disusun oleh Ir. Rinaldi Munir, M.T., dijelaskan bahwa inti algoritma Breadth First Search adalah:

- Traversal dimulai dari simpul v .
- Algoritma:
 1. Kunjungi simpul v ,
 2. Kunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu.
 3. Kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya.
- Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi



Gambar 1. Graf tak berarah

Apabila pada graf di atas digunakan algoritma Breadth First Search, maka pengunjungan simpul-simpul akan dilakukan seperti berikut:

- Simpul awal 0, urutan simpul yang dikunjungi adalah 0,1,2,3,5,4,6,7
- Simpul awal 1, urutan simpul yang dikunjungi adalah 1,0,2,3,5,4,6,7
- Simpul awal 2, urutan simpul yang dikunjungi adalah 2,0,1,3,5,4,6,7
- Simpul awal 3, urutan simpul yang dikunjungi adalah 3,2,4,6,0,1,5,7
- Simpul awal 4, urutan simpul yang dikunjungi adalah 4,3,6,2,0,1,5,7
- Simpul awal 5, urutan simpul yang dikunjungi adalah 5,2,7,0,1,3,4,6
- Simpul awal 6, urutan simpul yang dikunjungi adalah 6,3,4,2,0,1,5,7
- Simpul awal 7, urutan simpul yang dikunjungi adalah 7,5,2,0,1,3,4,6

Jelas bahwa semua simpul akan dikunjungi, tetapi prioritas kunjungan diberikan kepada simpul yang paling dekat dengan simpul pertama yang telah ditetapkan.

3. Pengembangan Algoritma Breadth First Search dalam Aplikasi Penduplikasian Situs

Dalam pengembangannya tentu saja diperlukan algoritma yang efektif dan efisien sehingga penduplikasian situs bisa berjalan dengan mudah.

Algoritma Breadth First Search yang telah dikembangkan menjadi algoritma untuk menduplikasi situs selengkapnya adalah:

```

procedure DuplikasiSitus(input
v:link, n:integer)
{ Traversal link yang terdapat dalam
situs dengan algoritma BFS.
Masukan: v adalah link alamat
situs pertama yang akan dikunjungi
n adalah kedalaman link
yang akan dikunjungi program

```

Keluaran: semua link sampai kedalaman ke-n akan dikunjungi dan disimpan }

Deklarasi

```

level : integer
l : link
antrian[1..n]: antrianLink
{deklarasi antrian berdasarkan
level}
df : daftarDownloadedLink

```

```

procedure BuatAntrian(input/output
q : antrianLink
{ membuat antrian kosong, kepala(q)
diisi 0 }

```

```

procedure MasukAntrian(input/output
q:antrianLink, input l:link)
{ membaca seluruh link yang
terdapat dalam link l dan
menggabungkannya ke dalam antrian q
}

```

```

procedure HapusAntrian(input/output
q:antrianLink, output l:link)
{menghapus l dari kepala antrian q}

```

```

procedure simpanAlamat(input/output
df:daftarDownloadedLink, input
l:link)
{ menambah daftar downloaded file
df dengan alamat l karena l sudah
dikunjungi }

```

```

function AntrianKosong(input
q:antrian) → boolean
{ true jika antrian q kosong, false
jika sebaliknya }

```

```

function checkLink (input
df:daftarDownloadedLink, l:link) →
boolean
{ true jika link l terdapat dalam
df, false jika sebaliknya }

```

Algoritma:

```

for i=1 to n do
  BuatAntrian(antrian[i])
end for

```

```

BuatDaftarDownloadedFile(df) {
  buat daftar-file kosong }

```

```

openLink(v)
{ membuka link ke situs utama }

```

```

saveLink(v)
{ menyimpan halaman situs v dalam
bentuk web archive/file tunggal }

```

```

simpanAlamat(df,v)
{ alamat situs v disimpan }

MasukAntrian(antrian[1],1)
{ masukkan simpul awal kunjungan ke
dalam antrian}

{ mulai mengunjungi link-link yang
terdapat dalam file }

for i=1 to n do
  while not
antrianKosong(antrian[i]) do
  hapusAntrian (antrian[i],1)
  if not checkLink(df,1) then
    openLink(1)
    saveLink(1)
    simpanAlamat(df,1)
    masukAntrian(antrian[i+1],1)
  endif
endwhile
endfor

```

2. <http://www.seas.gwu.edu/~simhaweb/cs151/lectures/module7/module7.html> diakses tanggal 17 Mei 2006

Algoritma diatas apabila dikembangkan lagi dapat dipergunakan untuk membuat sebuah aplikasi yang dapat dipergunakan untuk menduplikasi situs.

4. Kompleksitas

Pada dasarnya keseluruhan ide algoritma di atas hampir sama dengan ide algoritma BFS, sehingga perhitungan kompleksitasnya sama dengan algoritma BFS, hanya dalam algoritma di atas ada fungsi/prosedur yang menambah kompleksitas algoritma, contohnya: fungsi checkLink.

Kompleksitas memang menentukan dalam berapa lama aplikasi dengan algoritma tersebut berjalan, tetapi yang lebih berpengaruh lagi adalah kecepatan jaringan dan kinerja komputer yang menggunakan algoritma ini.

5. Kesimpulan

Penggunaan algoritma Breadth First Search dalam aplikasi penduplikasian situs akan sangat menghemat waktu, karena akan menghemat waktu delay antara membuka link pertama-menyimpannya dengan membuka link kedua-menyimpannya.

Algoritma Breadth First Search akan mengunjungi setiap link yang terdapat dalam situs utama sampai kedalaman ke-n tanpa terlupa, karena algoritma ini adalah salah satu algoritma berjenis traversal.

Daftar Pustaka

1. Rinaldi Munir, *Diktat Kuliah IF2251 Strategi Algoritmik*, Bandung, 2005.