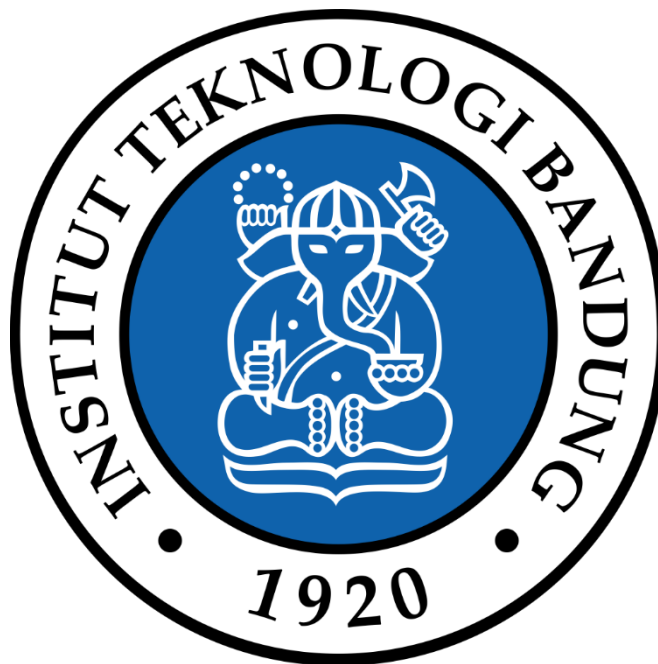


LAPORAN TUGAS BESAR 2
IF2211 STRATEGI ALGORITMA
PENGAPLIKASIAN ALGORITME BFS DAN DFS DALAM
IMPLEMENTASI FOLDER CRAWLING

Oleh
IGNATIUS DAVID PARTOGI
NIM : 13518014



PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
Maret 2022

1. Penjelasan Algoritme BFS dan DFS

Pada program ini, definisi graph dibuat menggunakan dictionary, dengan key yang menandakan node dan value yang menandakan tetangga-tetangga yang dimiliki.

a. Algoritme BFS

Pada program ini, algoritme BFS bekerja menggunakan sistem queue. Pada awal eksekusi, dilakukan loop while dengan break point berupa jika isi queue kosong dan dilakukan pencarian dalam loop tersebut. Element pertama pada queue (selalu root) dikeluarkan dari queue dan digunakan sebagai path untuk pencarian. Jika tidak ditemukan nama file yang diinginkan pada directory tersebut, program akan memeriksa apakah directory tersebut memiliki tetangga atau subdirectory di dalamnya. Jika ada, maka semua path subdirectory tersebut dimasukkan ke bagian belakang queue secara terurut dan loop akan berlanjut dengan memeriksa elemen pertama pada queue tersebut sampai file ditemukan (untuk pencarian satu file) atau semua directory di cek (untuk pencarian semua file dengan nama yang sama).

b. Algoritme DFS

Algoritme DFS pada program ini memiliki cara kerja yang serupa dengan algoritme BFS, namun menggunakan sistem stack, bukan queue. Pada awal eksekusi, dilakukan loop while dengan break point berupa jika isi stack kosong dan dilakukan pencarian dalam loop tersebut. Element pertama pada stack (selalu root) dikeluarkan dari stack dan digunakan sebagai path untuk pencarian. Jika tidak ditemukan nama file yang diinginkan pada directory tersebut, program akan memeriksa apakah directory tersebut memiliki tetangga atau subdirectory di dalamnya. Jika ada, maka semua path subdirectory tersebut akan di balik (contoh, jika semua subdirectory dari sebuah directory secara terurut adalah 1, 2, 3, 4, maka akan dibalik menjadi 4, 3, 2, 1), dimasukkan ke bagian depan stack secara iteratif mulai dari subdirectory pertama, dan loop akan berlanjut dengan memeriksa elemen pertama pada stack tersebut sampai file ditemukan (untuk pencarian satu file) atau semua directory di cek (untuk pencarian semua file dengan nama yang sama).

2. Input/Output Program

Program meminta 4 macam input, yaitu:

1. Path folder untuk memulai pencarian
2. Nama file
3. Apakah pencarian hanya untuk satu file atau semua file dengan nama yang sama
4. Algoritme pencarian (BFS atau DFS)

Program mengeluarkan 3 macam output:

1. Hyperlink dari path file yang merupakan file yang dicari
2. Waktu eksekusi
3. Graf dengan atribut:
 - Kotak node dan panah berwarna hitam menandakan node tersebut berada pada antrian namun belum dicari oleh program

- Kotak node dan panah berwarna merah menandakan node tersebut telah dicari oleh program
- Kotak node dan panah berwarna biru menandakan file berada pada directory tersebut (baik secara langsung maupun di dalam sub directory dari directory).

Form1

Change Folder... C:\Users\User\Documents\IF2211

LAPORAN TUGAS

☐ Find all occurrences

groupBox1

☒ BFS

☐ DFS

Search

Form1

<C:\Users\User\Documents\IF2211\Tucil 1\LAPORAN TUGAS KECIL 1.docx>

Change Folder... C:\Users\User\Documents\IF2211

JGAS KECIL 1.docx

☐ Find all occurrences

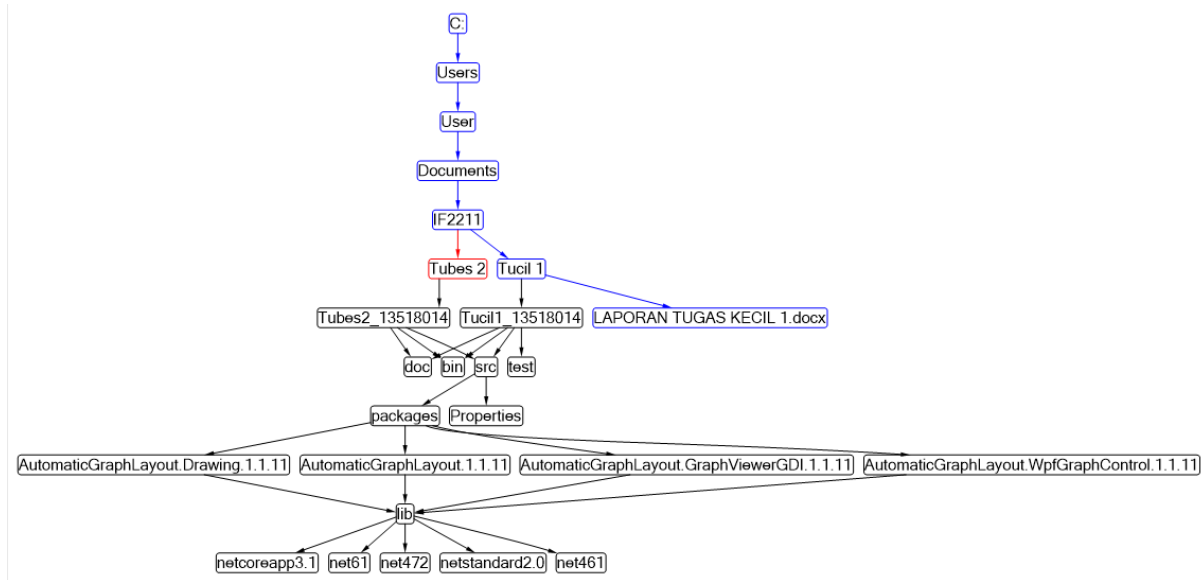
groupBox1

☒ BFS

☐ DFS

Search

Total Execution Time: 6ms



3. Repository menuju Source Code

<https://github.com/13518014Ignatius/Tubes-2-STIMA>