

LAPORAN TUGAS BESAR
MATA KULIAH STRUKTUR DATA

**“PENGUNAAN DEPTH FIRST SEARCH DALAM MENGIMPLEMENTASIKAN
PENCARIAN JALUR DI KABUPATEN BOGOR”**



Disusun Oleh :
Kelompok 3

NADINDA AURORA	121450004
DEDE MASITA	121450007
ANASTHASHYA RACHMAN	121450013
HARTITI FADILAH	121450031
NURUL KHOTIMAH	119160081

Dosen Pengampu :
Amalya Citra Pradana, S.Kom., M.Si., M.Sc.
Riksa Meidy Karim, S.Kom., M.Si., M.Sc.
Tirta Setiawan, S.Pd., M.Si.

PROGRAM STUDI SAINS DATA
JURUSAN SAINS
INSTITUT TEKNOLOGI SUMATERA
2022/2023

ABSTRAK

Perkembangan teknologi berarti perkembangan dari masa ke masa yang akan terus berkembang ke arah yang lebih canggih. Perkembangan ini didasarkan dari inovasi dan kreativitas manusia yang akan selalu membuat inovasi di berbagai bidang teknologi, salah satunya dari python yang merupakan bahasa pemrograman yang mampu membuat suatu aplikasi, serta dapat diterapkan sebagai pembuatan solusi untuk mencapai jalan yang efektif, penggunaan DFS(Depth First Search) yang di implementasikan terhadap jalan di kota Bogor melalui metode pencarian pada sebuah pohon dengan menelusuri satu cabang sebuah pohon sampai menemukan solusi. Pencarian akan dilakukan pada satu node dalam setiap level dari yang paling kiri dan dilanjutkan pada node sebelah kanan, hal ini dimaksudkan untuk mencapai rute tercepat sebagai solusi di daerah Indonesia yang sering terjadi kemacetan.

kata kunci: inovasi, DFS, solusi

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Semakin cepat perkembangan teknologi dari zaman ke zaman, semakin dituntut pula untuk setiap manusia dapat mengikuti perkembangan zaman. Hal ini sangat berkaitan dengan setiap orang untuk dapat mempercepat pekerjaannya dan diimbangi dengan efisien dan efektif dalam bekerja. Dengan adanya tuntutan pekerjaan begitu, mendorong manusia dalam menciptakan inovasi-inovasi dibidang teknologi yang dapat membantu manusia dalam menjalankan kehidupan. Kecanggihan atau inovasi yang akan diterapkan dapat didorong oleh kemajuan komputerisasi yang telah ada sebelumnya seperti menerapkan bahasa pemrograman python. Bahasa pemrograman python merupakan bahasa pemrograman yang dapat membuat suatu aplikasi, aplikasi yang dapat diterapkan seperti pembuatan solusi untuk mencapai jalan yang efektif yang akan dilalui dalam bekerja.

Program yang dapat membantu manusia adalah program melalui bahasa pemrograman python dengan metode Depth First Search yang berguna dalam mencari suatu solusi dengan aspek efisien, efektif, dan kecepatan. Kegunaan dalam metode Depth First Search sebagai meminimalisir adanya keterlambatan dalam bekerja. Dengan penerapan metode seperti ini, instansi dapat mengembangkan bukti konkret untuk pengambilan keputusan selanjutnya seperti pada pertimbangan alur yang digunakan, ketepatan waktu serta dapat mengefisienkan waktu yang digunakan. Program ini sangat dibutuhkan untuk menciptakan solusi terutama di daerah Indonesia yang sering terjadi kemacetan seperti di Kabupaten Bogor.

B. MASALAH

Dalam laporan ini yang berjudul “Penggunaan Depth First Search Dalam Pengimplementasian Pencarian Jalur Di Kabupaten Bogor” terdapat beberapa masalah yang harus dipecahkan, antara lain:

1. Bagaimana program pada python yang digunakan dalam Depth First Search (jalur tercepat) pada Pengimplementasian Pencarian Jalur Di Kabupaten Bogor?

2. Bagaimana sistem metode Depth First Search bekerja dalam pencarian jalur di Kabupaten Bogor?

C. DESKRIPSI DATA

Deskripsi data yang kami gunakan berupa DataSet dari Google Maps, yang dimana kami mengambil titik-titik penyusun graf yang berada disuatu wilayah, yaitu ada di Kota Bogor. Kami mengambil pusat di sekitar Kecamatan yang ada di Kota Bogor, yang berisikan nama-nama desa yang kami hubungkan menjadi suatu graf. Pada setiap tahapan proses DFS, dimulai dengan:

1. Penentuan v sebagai simpul,
2. Mengunjungi node v , kemudian node w , yang bersebelahan dengan node w ,
3. Ulangi DFS mulai dari node w ,
4. Ketika mencapai titik- u dengan semua titik tetangga yang dikunjungi, pencarian kembali ke node terakhir dikunjungi sebelumnya dan memiliki node yang belum dikunjungi w ,
5. Pencarian berakhir ketika tidak ada lagi node yang belum dikunjungi ditemukan dicapai oleh node yang dikunjungi.

Pada data ini menggunakan rincian sebagai berikut:

- Tanah Sareal memiliki akses ke Bogor Utara dan Bogor Barat,
- Bogor Utara memiliki akses ke Cimahpar dan Bantarjati,
- Bogor Barat,
- Bantarjati memiliki akses ke Bogor Tengah,
- Cimahpar memiliki akses ke Bogor Timur,
- Bogor Tengah memiliki akses ke Baranangsiang dan Pasirkuda,
- Pasirkuda memiliki akses ke Cikaret,
- Baranangsiang,
- Cikaret,
- Bogor Timur memiliki akses ke Sindangrasa dan Ranggamekar,
- Ranggamekar memiliki akses ke Bogor Selatan,
- Sindangrasa memiliki akses ke Harjasari,
- Bogor Selatan memiliki akses ke Kertamaya dan Pamoyaman,
- Harjasari,
- Pamoyaman,
- Kertamaya.

Lebih jelasnya terlampir pada graf sederhana yang kami ambil melalui gambar yang berasal dari Google dibawah ini:



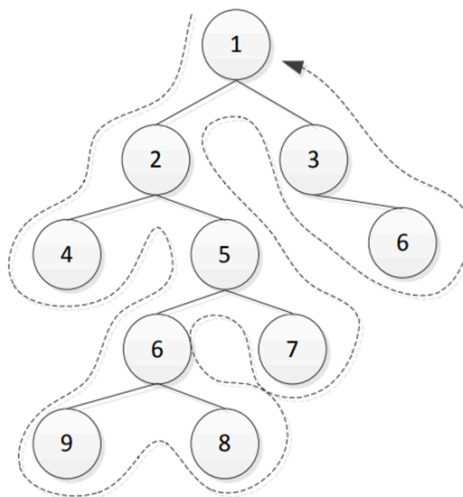
Gambar 1. Peta kabupaten bogor

BAB II

RANCANGAN SISTEM

A. GAMBARAN UMUM

Algoritma *Depth-First-Search* atau disingkat dengan DFS adalah algoritma pencarian yang akan melakukan ekspansi node atau child pertama (Freeman,1991). Sehingga DFS akan melakukan pencarian ke kedalaman maksimal dari sebuah tree atau graph. Algoritma ini akan bersifat complete apabila graph atau tree memiliki sifat finite state space namun apabila infinite state space, maka algoritma tidak complete. Menurut Inggiantowi (2011), *Depth first search* (DFS) adalah algoritma umum traversal atau pencarian pada pohon, struktur pohon, atau graf .



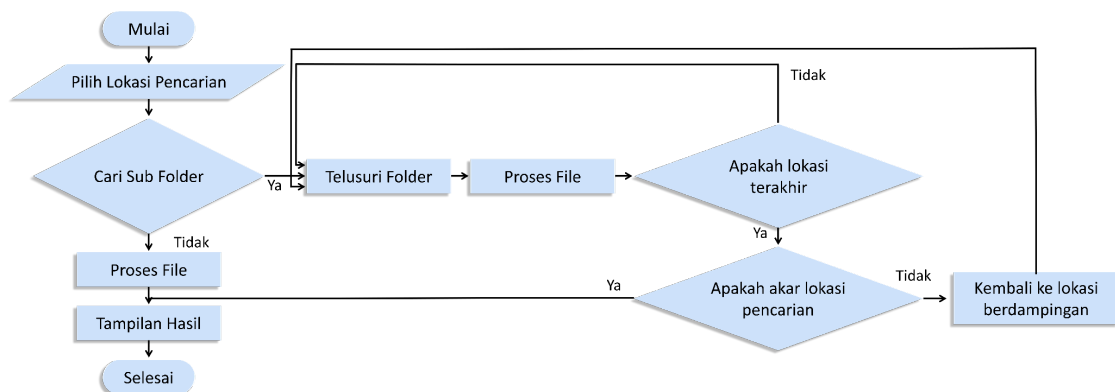
Gambar 2. Graf DFS

Pada pohon pencarian Gambar graf *Depth first search*, dianggap titik asal adalah lingkaran satu. Seluruh lintasan memiliki hubungan satu arah. Semua lintasan hanya berjalan satu arah dari atas kebawah, dimulai dari simpul akar dan mengecek simpul anaknya yang pertama, setelah itu, algoritma mengecek simpul anak dari simpul anak yang pertama tersebut, hingga mencapai simpul daun atau simpul tujuan. Jika solusi belum ditemukan, algoritma melakukan runut balik (backtracking) ke simpul orangtuanya yang paling baru diperiksa lalu dan mengecek simpul anaknya yang belum diperiksa, sedemikian seterusnya hingga simpul solusi ditemukan (Hendry,2011). Lingkaran yang memiliki dua simpul berarti dari lingkaran tersebut bisa menuju ke dua tujuan yaitu lingkaran disimpulnya. Keseluruhan dari lingkaran pada data akan dicek apakah memiliki simpul atau tidak. Apabila menemukan titik

tujuan atau sebuah lingkaran tidak memiliki simpul, maka algoritma akan kembali ke simpul asalnya. Hingga kembali ke titik pertama atau titik asal. Algoritma juga bisa diberikan kondisi khusus untuk menghentikan pencariannya.

Depth First Search memiliki beberapa kelebihan diantaranya yaitu cepat mencapai kedalaman ruang pencarian. Jika diketahui bahwa lintasan solusi permasalahan akan panjang maka *Depth first search* tidak akan memboroskan waktu untuk melakukan sejumlah besar keadaan dangkal dalam permasalahan graf. *Depth First Search* jauh lebih efisien untuk ruang pencarian dengan banyak cabang karena tidak perlu mengeksekusi semua simpul pada suatu level tertentu pada daftar open. Selain itu, *Depth First Search* memerlukan memori yang relatif kecil karena banyak node pada lintasan yang aktif saja yang disimpan. Selain kelebihan, *Depth First Search* juga memiliki kelemahan diantaranya adalah memungkinkan tidak ditemukannya tujuan yang diharapkan dan hanya akan mendapatkan satu solusi pada setiap pencarian.

B. FLOWCHART PENCARIAN JALUR DI KABUPATEN BOGOR

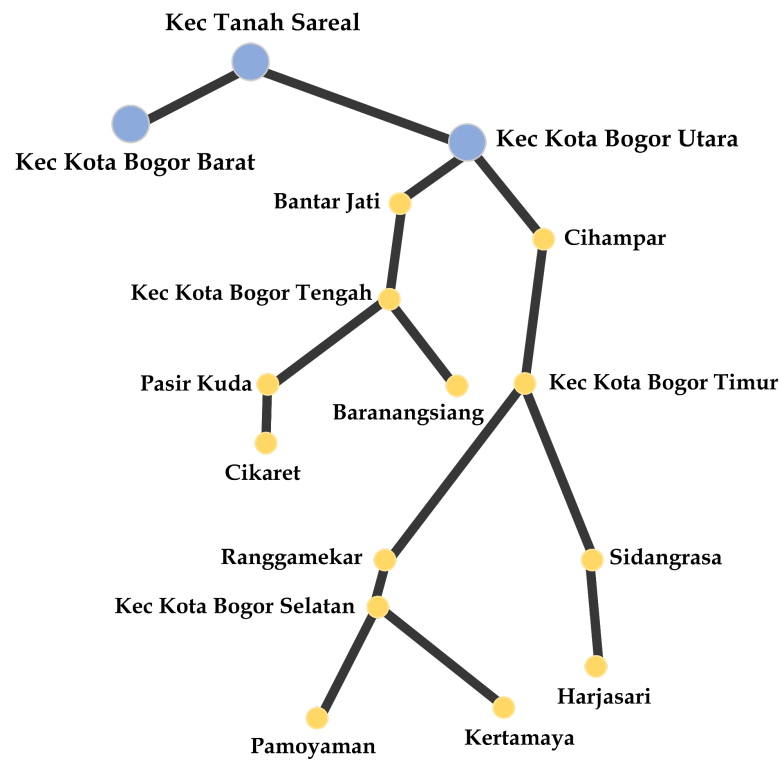


Gambar 3. Flowchart

Pada gambar diatas merupakan flowchart dimana proses diawali dengan pemilihan pencarian apakah pencarian cukup di folder akar atau pilihan ke sub folder. Jika hanya di folder akar maka dilanjutkan langsung dengan pemrosesan file teks dan menampilkan hasilnya. Dan jika memilih pencarian ke sub folder maka dilakukan penelusuran lokasi, yaitu pemeriksaan kepada setiap folder hingga didapatkan folder terakhir. Pada saat posisi di folder terakhir inilah dilakukan pemrosesan file yaitu pencarian kemiripan kata antara kata kunci dan kata pada dokumen, dilanjutkan

pencarian kembali pada lokasi berdampingan. Proses pencarian berhenti ketika lokasi folder telah berada pada akar lokasi.

Dengan algoritma *Depth First Search* berdasarkan pengimplementasian pada pencarian jalur kabupaten bogor maka dapat digambarkan sebagai bentuk graf berikut



C. DEPENDENCY / LIBRARY YANG DIGUNAKAN

Pada Tugas Besar yang kami ambil tentang **“Penggunaan Depth First Search Dalam Pengimplementasian Pencarian Jalur di Kabupaten Bogor”** tidak menggunakan library karena kami menginputkan data secara manual.

D. DAFTAR FUNGSIONALITAS / FITUR

Fungsionalitas adalah sesuatu yang berisi tentang proses-proses apa saja yang nantinya akan dilakukan oleh sistem. Fungsionalitas berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem.

Berikut ini adalah daftar fungsionalitas dari **“Penggunaan Depth First Search Dalam Pengimplementasian Pencarian Jalur di Kabupaten Bogor”**

1. `graph{}`
`graph{}` berfungsi untuk membuat graf dengan menggunakan *list* untuk menyimpan titik yang menjadi tetangga sebuah titik.
2. **`def dfsPath(graph,start,goal):`**
Berfungsi untuk mencari semua jalur yang melalui titik.
3. **`stack = [[start]]`**
Stack biasa digunakan dalam mengontrol operasi dalam sebuah sistem operasi.
4. **`while stack:`**
While adalah salah satu pernyataan yang berfungsi untuk mengulangi pengeksekusian *sub statement* yang dilakukan ketika memiliki nilai benar pada *conditional expression*.
5. `append(child)`
Berfungsi untuk menambahkan sebuah elemen baru tanpa harus menyertakan element tersebut.

BAB III

HASIL PROGRAM

```
graph1 = {'Tanah_Sareal':(['Bogor_Utara', 'Bogor_Barat']),
          'Bogor_Utara':(['Cimahpar', 'Bantarjati']),
          'Bogor_Barat':([],),
          'Bantarjati':(['Bogor_Tengah']),
          'Cimahpar':(['Bogor_Timur']),
          'Bogor_Tengah':(['Baranangsiang', 'Pasirkuda']),
          'Pasirkuda':(['Cikaret']),
          'Baranangsiang':([],),
          'Cikaret':([],),
          'Bogor_Timur':(['Sindangrasa', 'Ranggamekar']),
          'Ranggamekar':(['Bogor_Selatan']),
          'Sindangrasa':(['Harjasari']),
          'Bogor_Selatan':(['Kertamaya', 'Pamoyaman']),
          'Harjasari':([],),
          'Pamoyaman':([],),
          'Kertamaya':([],)}
```

Pada *code* diatas dilakukan deklarasi objek dengan variabel *graph1* dan objek lainnya yang saling terhubung. Pada *code* yang terletak didalam kurung kurawa menandakan bahwa, *node* yang saling terhubung dan pada *node* yang memiliki list kosong menandakan bahwa, tidak terdapat *node* lain yang terhubung dengannya.

```
def dfsPath(graph,start,goal):
    stack = [[start]]
    visited = []

    while stack:
        print("stack", stack)
        path= stack.pop(0)
        node = path [-1]
        if node == goal:
            return print("The path is :", path)

        children = graph[node]
        for child in children:
            if child not in visited:
                newPath = path + [child]
                stack.insert(0, newPath)
                visited.append(child)
```

```
dfsPath (graph1,'Tanah_Sareal','Kertamaya')
```

Pada *code* diatas, membuat fungsi menggunakan def dan dfsPath. dfsPath merupakan implementasi dari stack sehingga, pada code diatas menggunakan stack dengan prinsip LIFO (Last In First Out). Terdapat juga code untuk menampilkan path (jalur) dengan stack.pop(0) yang berarti menghapus elemen pertama. Dilakukan deklarasi untuk newPath yang menampilkan path dan child nya. Pada baris terakhir dilakukan deklarasi untuk menampilkan output dengan format nama graph, asal, tujuan. Sehingga, output yang dihasilkan sebagai berikut:

```
stack [['Tanah_Sareal']]
stack [['Tanah_Sareal', 'Bogor_Barat'], ['Tanah_Sareal', 'Bogor_Utara']]
stack [['Tanah_Sareal', 'Bogor_Utara']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati', 'Bogor_Tengah'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati', 'Bogor_Tengah', 'Pasirkuda'], ['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati', 'Bogor_Tengah', 'Baranangsiang'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati', 'Bogor_Tengah', 'Pasirkuda', 'Cikaret'], ['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati', 'Bogor_Tengah', 'Baranangsiang'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Bantarjati', 'Bogor_Tengah', 'Baranangsiang'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Ranggamekar'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Sindangrasa']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Ranggamekar', 'Bogor_Selatan'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Sindangrasa']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Ranggamekar', 'Bogor_Selatan', 'Pamoyaman'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Ranggamekar', 'Bogor_Selatan', 'Kertamaya'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Sindangrasa']]
stack [['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Ranggamekar', 'Bogor_Selatan', 'Kertamaya'], ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Sindangrasa']]
The path is : ['Tanah_Sareal', 'Bogor_Utara', 'Cimahpar', 'Bogor_Timur', 'Ranggamekar', 'Bogor_Selatan', 'Kertamaya']
```

Pada output diatas, didapatkan jalur tercepat yang dijelaskan oleh The path is dan metode Depth First Search dalam mencari jalur tercepat.

BAB IV

KESIMPULAN

Algoritma yang digunakan sebagai acuan dalam menentukan rute terbaik (tercepat) untuk sampai pada tujuan sesuai perintah yang dimasukkan, dari data yang digunakan yaitu data dari rute Kota Bogor yang berfokus pada jalanan mulai dari berbagai kecamatan hingga desa-desa yang ada di Kota Bogor. Algoritma yang diterapkan saat ini merupakan algoritma Depth First Search (DFS).

Dengan metode DFS yang kelompok kami terapkan di Kota Bogor, mulai dari Kecamatan Tanah Sareal hingga Desa Kertamaya, didapatkan bahwa rute terbaik adalah Kecamatan Tanah Sareal - Kecamatan Bogor Utara - Desa Cimahpar - Kecamatan Bogor Timur - Desa Ranggamekar - Kecamatan Bogor Selatan - Desa Kertamaya. Kelebihan penggunaan DFS dalam penentuan rute perjalanan ini adalah kita dapat mengetahui seluruh jalan-jalan yang dapat kita lewati dengan penerapan fungsi stack, terdapat pula kekurangan pada penggunaan DFS ini yaitu hasil pencarian rute yang akan selalu bekerja mulai dari kiri pohon (tree) yang terbentuk, tetapi pada percobaan kali ini siklus yang diterapkan adalah penentuan rute dari suatu tempat ke tempat lainnya mengikuti pohon (tree) yang terbentuk.

SARAN

Adapun beberapa saran yang dapat bermanfaat sebagai arahan dan membangun pada praktikum selanjutnya, antara lain:

1. Diharapkan untuk pengembangan selanjutnya dapat menerapkan algoritma lain selain DFS.
2. Diharapkan untuk pengembangan selanjutnya tampilannya dibuat lebih menarik.
3. Diharapkan DFS yang dibuat dapat diterapkan dalam sehari - hari.

BAB V

REFERENSI

Jon Freeman, "Parallel Algorithms for Depth-First Search", University of Pennsylvania Departement of Computer and Information Science Technical Report No. MS-CIS-91-71, 1991

Hendry. 2011. Perbandingan Metode Depth First Search (DFS) dan Breadth First Search (BFS) untuk Mengidentifikasi Kerusakan Handphone. Medan: STMIK IBBI

Inggiantowi,Hafid.2008. Perbandingan Algoritma Penelusuran Depth First Search dan Breadth First Search pada Graf serta Aplikasinya. Bandung: ITB