

Nama: Dewa Bagus Putu Arya Dhananjaya

NPM: 10122362

Kelas: 3KA21

Implementasi Algoritma Depth First Search (DFS) dengan Python

1. Definisi Algoritma Depth-First Search (DFS)

Apa itu DFS? Depth-First Search (DFS) adalah salah satu algoritma pencarian dalam struktur data graf atau pohon. Algoritma ini bekerja dengan menelusuri simpul (node) secara mendalam sebelum kembali ke simpul sebelumnya untuk menjelajahi cabang lainnya.

DFS menggunakan pendekatan rekursif atau menggunakan struktur data stack untuk mengelola urutan eksplorasi simpul.

DFS sering digunakan dalam berbagai aplikasi, seperti:

- Pencarian jalur dalam graf (pathfinding)
- Analisis hubungan antar data (misalnya, sistem rekomendasi)
- Menemukan komponen terhubung dalam graf
- Penyelesaian teka-teki seperti maze solving

2. Cara Implementasi Algoritma DFS dalam Program Python

Struktur Program

Program ini terdiri dari tiga bagian utama:

- Representasi Graf → menggunakan dictionary Python
- Fungsi DFS → menggunakan pendekatan rekursi
- Menjalankan DFS dari simpul awal

(A) Representasi Graf Menggunakan Dictionary

Di dalam kode, graf direpresentasikan menggunakan struktur data dictionary. Setiap kunci (key) merepresentasikan sebuah simpul (node), sedangkan nilai (value) adalah daftar simpul yang terhubung dengannya.

```
In [1]: graph = {
    'A' : ['Z', 'S', 'T'],
    'Z' : ['O'],
    'O' : ['S1'],
    'S1' : ['F', 'R'],
    'F' : ['B'],
    'B' : [],
    'R' : [],
    'S' : ['O1', 'F1', 'R1'],
    'O1' : [],
    'F1' : ['B1'],
    'B1' : [],
    'R1' : ['C', 'P'],
    'C' : [],
    'P' : [],
    'T' : ['L'],
    'L' : ['M'],
    'M' : ['D'],
    'D' : []
}
```

Penjelasan Struktur Graf:

- Simpul 'A' memiliki tiga tetangga: 'Z', 'S', dan 'T'.
- Simpul 'Z' hanya terhubung ke 'O'.
- Simpul 'O' terhubung ke 'S1', yang kemudian bercabang ke 'F' dan 'R'.
- Setiap simpul yang tidak memiliki tetangga direpresentasikan sebagai daftar kosong [], seperti 'B', 'R', 'O1', dll.

Graf ini memiliki cabang-cabang yang memungkinkan DFS menelusuri jalur yang panjang sebelum kembali ke simpul awal.

(B) Implementasi Fungsi DFS dengan Rekursi

```
In [2]: visited = set() # Set to keep track of visited nodes of graph.
```

```
In [3]: def dfs(visited, graph, node): #function for dfs
    if node not in visited:
        print (node)
        visited.add(node)
        for neighbour in graph[node]:
            dfs(visited, graph, neighbour)
```

Cara Kerja Fungsi DFS:

1. Mengecek apakah simpul sudah dikunjungi

-Jika node belum ada di dalam visited, maka program akan mencetaknya dan menandainya sebagai sudah dikunjungi.

2. Menelusuri simpul yang terhubung

-Program akan melakukan loop untuk setiap tetangga yang ada dalam graph[node], lalu memanggil fungsi dfs() secara rekursif.

-Dengan pendekatan ini, program akan menjelajahi satu cabang hingga mencapai simpul yang tidak memiliki tetangga sebelum kembali ke simpul sebelumnya.

(C) Menjalankan DFS dari Simpul 'A'

```
In [4]: print("Following is the Depth-First Search")
```

```
Following is the Depth-First Search
```

```
In [5]: dfs(visited, graph, 'A')
```

```
A
Z
O
S1
F
B
R
S
O1
F1
B1
R1
C
P
T
L
M
D
```

-Program akan mencetak urutan DFS mulai dari simpul 'A' dan menjelajahi simpul yang terhubung secara rekursif.

-Hasil eksplorasi DFS bergantung pada urutan daftar tetangga dalam dictionary graph.

3. Kesimpulan:

- DFS adalah algoritma penelusuran graf yang bekerja dengan menelusuri simpul secara mendalam sebelum kembali ke simpul sebelumnya.
- Graf direpresentasikan sebagai dictionary di mana setiap simpul memiliki daftar tetangga yang terhubung dengannya.
- Fungsi DFS menggunakan rekursi untuk mengeksplorasi setiap simpul dan menandai yang sudah dikunjungi agar tidak dikunjungi dua kali.
- Output menunjukkan urutan kunjungan simpul sesuai dengan metode DFS, yaitu dengan menelusuri satu cabang hingga akhir sebelum kembali ke simpul sebelumnya.