

LAPORAN TUGAS KECIL 2 STRATEGI ALGORITMA

Penyusunan Rencana Kuliah dengan Topological Sort (Decrease and Conquer)



Oleh :

Made Kharisma Jagaddhita (13519176)

**Institut Teknologi Bandung
2021**

A. *Topological Sort*

Topological sort adalah pengurutan simpul-simpul dari graf berarah sedemikian rupa sehingga untuk setiap garis penghubung $u-v$ dari simpul u ke simpul v , simpul u berada sebelum simpul v dalam pengurutannya. *Topological sort* hanya dapat dilakukan jika graf yang ingin diurutkan merupakan *Directed Acyclic Graph* (DAG), yaitu graf yang tidak memiliki siklus. Hasil pengurutan dari *topological sort* untuk satu DAG dapat berbeda (tidak unik).

Berikut merupakan cara kerja algoritma dari *topological sort*:

1. Hitung semua derajat-masuk (in-degree) setiap simpul, yaitu banyaknya busur yang masuk pada simpul tersebut.
2. Pilih sembarang simpul yang memiliki derajat-masuk 0.
3. Ambil simpul tersebut, dan hilangkan simpul tersebut beserta semua busur yang keluar dari simpul tersebut pada graf, dan kurangi derajat simpul yang berhubungan dengan simpul tersebut dengan 1.
4. Ulangi langkah (2) dan (3) hingga semua simpul pada DAG terpilih.

Topological sorting merupakan implementasi dari algoritma *Decrease and Conquer*. *Decrease and Conquer* adalah perancangan algoritma dengan mereduksi persoalan menjadi dua upa-persoalan (sub-problem) yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Dalam *topological sorting* persoalan direduksi menjadi dua upa-persoalan. Satu upa-persoalan merupakan simpul yang memiliki derajat-masuk 0 yang merupakan solusi, sedangkan upa-persoalan lainnya merupakan sebuah graf yang simpul berderajat-masuk 0-nya "direduksi". Upa-persoalan yang berbentuk graf akan direduksi terus-menerus hingga tidak lagi terdapat simpul (*Decrease*). Kemudian solusi-solusi dari setiap upa-persoalan akan digabungkan sehingga membentuk *topological ordering* (*Conquer*).

B. Source code

Program memakai bahasa python dan diberi nama matkul.py. Source code dapat diakses di link ini: https://github.com/kharismajd/Tucil2_13519176

```
1  ### Fungsi-fungsi tambahan ###
2  def getProbleml(fileName) :
3      # Mengembalikan matrix dari file fileName
4      # Contoh: isi file.txt:
5      # C1, C3.
6      # C2, C1, C4.
7      # C3.
8      # C4, C1, C3.
9      # C5, C2, C4.
10     # Maka akan mengembalikan:
11     # [
12     #     ["C1", "C3"],
13     #     ["C2", "C1", "C4"],
14     #     ["C3"],
15     #     ["C4", "C1", "C3"],
16     #     ["C5", "C2", "C4"]
17     # ]
18     f = open("../test/" + fileName, "r")
19     dag = []
20     for line in f :
21         line = line.replace(".", "")
22         line = line.replace(" ", "")
23         line = line.replace("\n", "")
24         line = line.split(",")
25         dag.append(line)
26     f.close()
27     return dag
28
```

```
29 def find_zero_in_degree(dag) :
30     # Mengembalikan array simpul simpul yang
31     # memiliki in-degree 0
32     # Contoh: dag =
33     # [
34     #     ["C2"],
35     #     ["C3"],
36     #     ["C4", "C1", "C2", "C3"],
37     #     ["C5", "C1", "C3", "C4"]
38     # ]
39     # Maka akan mengembalikan:
40     # ["C2", "C3"]
41     zero_in_degree = []
42     for i in range(len(dag)) :
43         if (len(dag[i]) == 1) :
44             zero_in_degree.append(dag[i][0])
45     return zero_in_degree
46
```

```

47 ▾ def delete_dag_node(dag, nodes) :
48     # Menghapus simpul-simpul dan busur yang keluar dari
49     # simpul tersebut di dag yang ada di array nodes
50     # Contoh:
51     # dag =
52     # [
53     #     ["C2"],
54     #     ["C3"],
55     #     ["C4", "C2", "C3"],
56     #     ["C5", "C3", "C4"]
57     # ]
58     # nodes = ["C2", "C3"]
59     #
60     # Maka dag akan berubah menjadi:
61     # dag =
62     # [
63     #     ["C4"],
64     #     ["C5", "C4"]
65     # ]

```

```

66     empty = False
67     i = 0
68     while ((not(empty)) and (i < len(dag))) :
69         j = 0
70         while ((not(empty)) and (j < len(dag[i]))) :
71             found = False
72             k = 0
73             while ((not(found)) and (k < len(nodes))) :
74                 if (dag[i][j] == nodes[k]) :
75                     dag[i].pop(j)
76                     found = True
77                     j -= 1
78                     k += 1
79                 j += 1
80             if (len(dag[i]) == 0) :
81                 dag.pop(i)
82                 i -= 1
83             if (len(dag) == 0) :
84                 empty = True
85             i += 1
86     return 0
87

```

```

88 def topo_sort(dag) :
89     # Mengembalikan simpul hasil sorting dag
90     # Jika tidak ada solusi, maka akan mengembalikan 0
91     # Contoh:
92     # dag =
93     # [
94     #     ["C1"],
95     #     ["C2", "C1"],
96     #     ["C3", "C1"],
97     #     ["C4", "C1", "C2", "C3"],
98     #     ["C5", "C1", "C3", "C4"]
99     # ]
100     # Maka akan mengembalikan
101     # [["C1"], ["C2", "C3"], ["C4"], ["C5"]]
102     solution = []
103     dag_copy = [node[:] for node in dag]    # Copy agar dag di parameter tidak terhapus
104     while(len(dag_copy) != 0) :
105         zero_in_degree = find_zero_in_degree(dag_copy)
106         if (len(zero_in_degree) > 0) :
107             solution.append(zero_in_degree)
108             delete_dag_node(dag_copy, zero_in_degree)
109         else :
110             return 0
111     return solution
112

```

```

113 ### Main program ###
114 dag = getProblem(input("Masukkan nama file: "))
115 solution = topo_sort(dag)
116 if (not(solution)) :
117     print("Graph bukan sebuah DAG")
118 else :
119     for i in range(len(solution)) :
120         print("Semester " + str(i + 1) + " : ", end="")
121         for j in range(len(solution[i])) :
122             if (j == 0) :
123                 print(solution[i][j], end="")
124             else :
125                 print(", " + solution[i][j], end="")
126         print()

```

C. Pengujian Program

Checklist pengujian:

Poin	Ya	Tidak
1. Program berhasil dikompilasi.	✓	
2. Program berhasil running.	✓	
3. Program dapat menerima berkas input dan menuliskan output.	✓	
4. Luaran sudah benar untuk semua kasus input.	✓	

Screenshot pengujian:

1. test1.txt :

C1, C3.
C2, C1, C4.
C3.
C4, C1, C3.
C5, C2, C4.

Hasil kompilasi:

```
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test1.txt
Semester 1 : C3
Semester 2 : C1
Semester 3 : C4
Semester 4 : C2
Semester 5 : C5
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> |
```

2. test2.txt:

C1.
C2, C1.
C3, C1.
C4, C1, C2, C3.
C5, C1, C3, C4.

Hasil kompilasi:

```

PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test2.txt
Semester 1 : C1
Semester 2 : C2, C3
Semester 3 : C4
Semester 4 : C5
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> 

```

3. test3.txt:

```

C1.
C2, C1.
C3, C1, C2.
C4, C3.
C5, C3.
C6, C5.
C7, C6, C10.
C8, C4, C7, C12.
C9, C5.
C10, C6, C9.
C11, C3.
C12, C11.
C13, C8.
C14, C8, C13, C15.
C15, C13.

```

Hasil kompilasi:

```

PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test3.txt
Semester 1 : C1
Semester 2 : C2
Semester 3 : C3
Semester 4 : C4, C5, C11
Semester 5 : C6, C9, C12
Semester 6 : C10
Semester 7 : C7
Semester 8 : C8
Semester 9 : C13
Semester 10 : C15
Semester 11 : C14
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py

```

4. test4.txt:

```

C0, C3, C7.
C1, C5, C7.
C2, C1.

```

C3, C7.
C4, C1, C3.
C5, C7.
C6, C0, C1.
C7.

Hasil kompilasi:

```
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test4.txt
Semester 1 : C7
Semester 2 : C3, C5
Semester 3 : C0, C1
Semester 4 : C2, C4, C6
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src>
```

5. test5.txt:

C1.
C2, C1.
C3, C1, C2.
C4, C2, C3.
C5, C3, C4.
C6, C4, C5.

Hasil kompilasi:

```
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test5.txt
Semester 1 : C1
Semester 2 : C2
Semester 3 : C3
Semester 4 : C4
Semester 5 : C5
Semester 6 : C6
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src>
```

6. test6.txt:

C2, C11.
C3.
C5.
C7.
C8, C7, C3.
C9, C8, C11.
C10, C3, C11.
C11, C5, C7.

Hasil kompilasi:


```

PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test6.txt
Semester 1 : C3, C5, C7
Semester 2 : C8, C11
Semester 3 : C2, C9, C10
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src>

```

7. test7.txt:

```

C1.
C2, C1.
C3, C2.
C4, C2.
C5, C3, C4.
C6, C5.
C7, C5.
C8, C3, C4, C6, C7.
C9, C8, C1.

```

Hasil kompilasi:

```

PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test7.txt
Semester 1 : C1
Semester 2 : C2
Semester 3 : C3, C4
Semester 4 : C5
Semester 5 : C6, C7
Semester 6 : C8
Semester 7 : C9
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src>

```

8. test8.txt:

```

C1, C2, C6, C8.
C2.
C3, C7.
C4, C1, C3, C7.
C5, C1, C6, C10.
C6, C11.
C7, C1, C11.
C8.
C9, C3.
C10.
C11.

```

Hasil kompilasi:

```
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> python matkul.py
Masukkan nama file: test8.txt
Semester 1 : C2, C8, C10, C11
Semester 2 : C6
Semester 3 : C1
Semester 4 : C5, C7
Semester 5 : C3
Semester 6 : C4, C9
PS C:\Users\Khari\Desktop\Semester 4\Stima\Tucil2_13519176\src> |
```