

**Penyusunan Rencana Kuliah dengan Topological Sort
(Penerapan Decrease and Conquer)**

LAPORAN TUGAS KECIL 2

MATA KULIAH IF2211 STRATEGI ALGORITMA



Disusun oleh: Aria Bachrul Ulum Berlian (13519115)

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Semester 2 Tahun 2020/2021**

Algoritma Topological Sort

Penyusunan rencana kuliah dengan penentuan mata kuliah mana yang terlebih dahulu harus diambil dapat dilakukan dengan algoritma topological sort. Mata kuliah prasyarat dan mata kuliah dapat direpresentasikan sebagai graf berarah. Simpul-simpulnya adalah mata kuliah. Ide utama dari topological sort adalah mencari item yang tidak memiliki predesesor. Item dalam kasus ini adalah mata kuliah dan predesesor adalah *prerequisitenya*. Tidak memiliki predesesor berarti derajat masuk sebuah simpulnya nol. Lalu, hapus item tersebut dari himpunan input dan masukkan ke dalam himpunan solusi. Selanjutnya, pencarian dilakukan lagi hingga himpunan input kosong. Pengurangan mata kuliah dari himpunan input ini sejalan dengan strategi algoritma decrease and conquer. Kita mereduksi persoalan menjadi dua sub persoalan dan hanya menggunakan satu sub persoalan saja. Lebih tepatnya, kita menggunakan algoritma decrease by constant, yakni mengurangi dengan 1.

Source Code Program

```
"""
PLANNER

Program Topological Sort untuk penyelesaian masalah pengambilan rencana kuliah
Asumsi : Kuliah dan pre-requisite nya berupa Directed Acyclic Graph (DAG)
"""

# sol    : array of solution
# graph  : array of tuple (pred, succ)

def read_file():

    # data    : matriks of data
    # data_perline : array of data per line

    graph = []
    data = []
    fn = input("Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'):")
    f = open(fn, "r")
    for line in f:
        data_perline = line.replace(".", "").rstrip()
        data_perline = data_perline.split(", ")
        data.append(data_perline)

    for i in range(len(data)):
        last = True
        for j in range(len(data)):
            if j == i:
                continue
            else:
                for k in range(len(data[j])):
                    if data[i][0] == data[j][k]:
                        graph.append((data[i][0], data[j][0]))
                        last = False
        if last:
            graph.append((data[i][0], None))

    f.close()
    return graph

def countInDegree(graph, el):
    count = 0
    for i in range(len(graph)):
        if graph[i][1] == el:
```

```

        count += 1
    return count

def deleteTuple(graph, el):
    i = 0
    while i < len(graph) and graph != []:
        if graph[i][0] == el:
            graph.pop(i)
            if len(graph) != 0:
                i -= 1
        else:
            i += 1
    return graph

def topo_sort(graph):
    sol = []
    j = 0
    need_to_delete = []
    while graph != []:
        i = 0
        sol.append(set())
        while i < len(graph):
            if countInDegree(graph, graph[i][0]) == 0:
                sol[j].add(graph[i][0])
                i += 1

        sol[j] = list(sol[j])
        for i in sol[j]:
            graph = deleteTuple(graph, sol[j][0])

        j += 1
    return sol

### Driver
g = read_file()
sol = topo_sort(g)

for i in range(len(sol)):
    print("Semester", i + 1, ":", end='')
    for j in range(len(sol[i])):
        if j != len(sol[i])-1:
            print(sol[i][j], ",", end='')
        else:
            print(sol[i][0])
    print('')

```

Pengujian

1.

```
tes.txt
1 C1, C3.
2 C2, C1, C4.
3 C3.
4 C4, C1, C3.
5 C5, C2, C4.

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 2: Code +
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"
Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes.txt
Semester 1 :C3

Semester 2 :C1

Semester 3 :C4

Semester 4 :C2

Semester 5 :C5
```

2.

```
tes1.txt
1 MK1.
2 MK2, MK1.
3 MK3, MK1.
4 MK4, MK1, MK2.
5 MK5, MK1, MK2, MK4, MK6.
6 MK6, MK1, MK3.
7 MK7, MK1, MK2, MK3, MK4, MK5, MK6.

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 1: Code +
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"
Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes1.txt
Semester 1 :MK1

Semester 2 :MK2 ,MK3

Semester 3 :MK4 ,MK6

Semester 4 :MK5

Semester 5 :MK7
```

3.

```
tes2.txt
1 MP0, MP4, MP5.
2 MP1, MP2, MP3, MP4, MP5.
3 MP2, MP5.
4 MP3, MP2, MP5.
5 MP4.
6 MP5.

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 1: Code +
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"
Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes2.txt
Semester 1 :MP4 ,MP5

Semester 2 :MP2 ,MP0

Semester 3 :MP3

Semester 4 :MP1
```

4.

```
tes3.txt
1 C1.
2 C2.
3 C3.
4 C4, C1, C2, C3.
5 C5, C1, C2, C3, C4.
6 C6, C1, C2, C3, C4.
7 C7, C1, C2, C3, C4.
8 C8, C1, C2, C3, C4, C5, C6, C7.
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 1: Code

Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes3.txt

Semester 1 :C3 ,C1 ,C2

Semester 2 :C4

Semester 3 :C7 ,C5 ,C6

Semester 4 :C8

5.

```
tes4.txt
1 CA.
2 CB, CA.
3 CC, CA.
4 CD, CA, CB, CC.
5 CE, CA, CB, CC, CD.
6
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 1: Code

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"

Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes4.txt

Semester 1 :CA

Semester 2 :CC ,CB

Semester 3 :CD

Semester 4 :CE

6.

```
tes5.txt
1 MPA.
2 MPB, MPA.
3 MPC, MPA, MPB.
4 MPD, MPA, MPB, MPC, MPG.
5 MPE, MPA, MPB, MPC, MPD.
6 MPF, MPA, MPB, MPC, MPD, MPE, MPG.
7 MPG.
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE 1: Code

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"

Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes5.txt

Semester 1 :MPA ,MPG

Semester 2 :MPB

Semester 3 :MPC

Semester 4 :MPD

Semester 5 :MPE

Semester 6 :MPF

7.

```
tes6.txt
1 database, oop.
2 oop, alstrukdat.
3 alstrukdat, daspro.
4 pengkom, kalkulus.
5 daspro, pengkom.
6 kalkulus.

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"
Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes6.txt
Semester 1 :kalkulus

Semester 2 :pengkom

Semester 3 :daspro

Semester 4 :alstrukdat

Semester 5 :oop

Semester 6 :database
```

8.

```
tes7.txt
1 MK5.
2 MK7.
3 MK3.
4 MK11, MK5, MK7.
5 MK8, MK7, MK3.
6 MK2, MK11, MK5.
7 MK9, MK8, MK7, MK3, MK11, MK7, MK5.
8 MK10, MK3, MK11, MK7, MK5.

D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2>python -u "d:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\planner.py"
Masukkan nama file (lengkap dengan alamat, cth:'D:\coba.txt'): D:\Kuliah\SMT 4\Strategi Algoritma\Tucil\2\tes7.txt
Semester 1 :MK5 ,MK3 ,MK7

Semester 2 :MK8 ,MK11

Semester 3 :MK10 ,MK9 ,MK2
```

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

SOURCE CODE: <https://github.com/ariaberlian/Planner>