

LAPORAN TUGAS KECIL 2
IF2211
STRATEGI ALGORITMA

Penyusunan Rencana Kuliah dengan *Topological Sort*



Disusun oleh:
Nabil Nabighah - 13519168

TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020/2021

A. Algoritma *Topological Sort*

1. Cari simpul dengan derajat masuk sebanyak nol di arraysimpul
2. Hapus simpul-simpul tersebut
3. Hapus semua sisi yang masuk dari simpul tersebut di simpul lain
4. Masukkan simpul yang nol tadi ke dalam array jawaban
5. Ulangi langkah 1 sampai 4 sampai seluruh array simpul kosong
6. Tampilkan simpul pada array jawaban

```
# Topological Sort
#ambil semua simpul yang nol degree innya
#delsisi
#delsimpul
#masukin simpul derajat nol ke array
def TopSort(answer):
    tempsimp = grf.getZeroInDegree()
    arrsimpul = []
    for i in tempsimp:
        simpulzero = i
        grf.delsisi(simpulzero)
        grf.delsimpul(simpulzero)
        arrsimpul.append(simpulzero)
    if len(arrsimpul) != 0:
        answer.append(arrsimpul)
```

B. Source Code

Note : Lebih lengkapnya lihat [disini](#)

```
1  #structure graf
2  class graf:
3      #constructor
4      def __init__(self,dictgraf = None):
5          #inisialisasi dictionary
6          if dictgraf is None:
7              dictgraf = {}
8          self.dictgraf = dictgraf
9
10     #METHOD
11
12     #METHOD SIMPUL
13
14     #add simpul
15     def addSimpul(self,simpul):
16         if simpul not in self.dictgraf:
17             self.dictgraf[simpul] = []
18
19     # delete simpul
20     def delSimpul(self,simpul):
21         if simpul in self.dictgraf:
22             del self.dictgraf[simpul]
23
24     # get banyak simpul
25     def getSimpul(self):
26         return len(self.dictgraf.keys())
27
28     # print Simpul
29     def printSimpul(self):
30         print(list(self.dictgraf.keys()))
31
32     # get derajat masuk
33     def getInDegree(self,simpul):
34         return len(self.dictgraf[simpul])
35
36     #ambil semua simpul berderajat 0 masukan ke dalam list
37     def getZeroInDegree(self):
38         templist = []
39         for simpul in self.dictgraf:
40             if len(self.dictgraf[simpul]) == 0:
41                 templist.append(simpul)
42         return templist
43
44     #METHOD SISI
45
46     #add sisi
47     def addSisi(self, sisi,simpul):
48         if simpul in self.dictgraf:
49             if len(self.dictgraf[simpul]) == 0:
50                 self.dictgraf[simpul] = [sisi]
51             else:
52                 self.dictgraf[simpul].append(sisi)
53
54     #delete sisi
55     def delSisi(self,sisi):
56         tempval = sisi
57         list2 = {simpul: [a for a in sisina if a not in tempval] for simpul,sisina in self.dictgraf.items()}
58         self.dictgraf = list2
59
60     #print sisi
61     def printSisi(self):
62         print(list(self.dictgraf.values()))
63
64
```

```

# Topological Sort
#ambil semua simpul yang nol degree innya
#delSisi
#delSimpul
#masukin simpul derajat nol ke array
def TopSort(answer):
    tempsimp = grf.getZeroInDegree()
    arrsimpul = []
    for i in tempsimp:
        simpulzero = i
        grf.delSisi(simpulzero)
        grf.delSimpul(simpulzero)
        arrsimpul.append(simpulzero)
    if len(arrsimpul) != 0:
        answer.append(arrsimpul)

#MAIN DRIVER
#inisialisasi array
p = []
p2 = []

#read file
#append ke list dengan formatnya[['C1','C3'],['C2','C1','C4']]
filename = input("enter filename (with extension): ")
with open(filename) as f:
    lines = [line.rstrip().replace(".",",").replace(" ",",")for line in f]
for item in lines:
    realitem = item.split(",")
    for k in realitem:
        if k != "":
            p2.append(k)
    p.append(p2)
    p2 = []

```

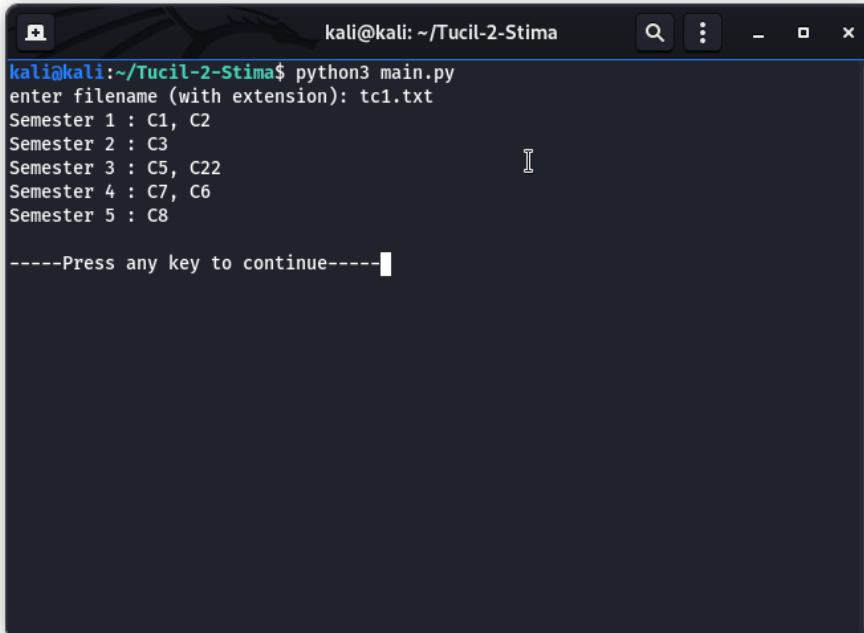
```

99
100
101 #inisiasi graf kosong
102 grf = graf()
103
104 #masukin semua simpul dan sisi ke graf
105 for array in range(len(p)):
106     grf.addSimpul(p[array][0])
107     tempsimpul = p[array][0]
108     del p[array][0]
109     for isiarray in range(len(p[array])):
110         grf.addSisi(p[array][isiarray],tempsimpul)
111
112 #Olah data memakai Topological Sort
113 answer = []
114 for i in range(8):
115     TopSort(answer)
116
117 #Handle yang ga bisa petakan
118 if grf.getSimpul() == 0:
119     for x in range(len(answer)):
120         print("Semester", x+1,":",end=" ")
121         for i in range(len(answer[x])):
122             if i == len(answer[x])-1:
123                 print(answer[x][i])
124             else:
125                 print(answer[x][i],end=" ")
126 else:
127     print("Tidak bisa menata mata kuliah")
128     print("Semester tidak cukup (>8 semester)")
129     print("")
130     input("-----Press any key to continue-----")

```

C. Screenshot

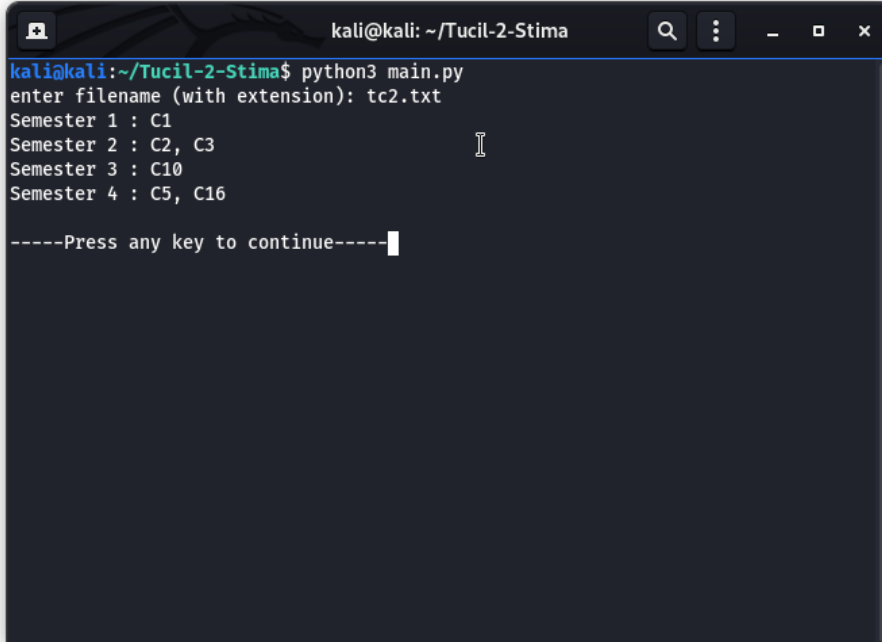
1. Test Case 1

A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The script prompts for a filename, then lists five semesters with their respective course numbers. Semester 1 has C1 and C2; Semester 2 has C3; Semester 3 has C5 and C22; Semester 4 has C7 and C6; Semester 5 has C8. The terminal ends with a prompt to press any key to continue.

```
kali@kali:~/Tucil-2-Stima$ python3 main.py
enter filename (with extension): tc1.txt
Semester 1 : C1, C2
Semester 2 : C3
Semester 3 : C5, C22
Semester 4 : C7, C6
Semester 5 : C8

-----Press any key to continue-----
```

2. Test Case 2

A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The script prompts for a filename, then lists four semesters with their respective course numbers. Semester 1 has C1; Semester 2 has C2 and C3; Semester 3 has C10; Semester 4 has C5 and C16. The terminal ends with a prompt to press any key to continue.

```
kali@kali:~/Tucil-2-Stima$ python3 main.py
enter filename (with extension): tc2.txt
Semester 1 : C1
Semester 2 : C2, C3
Semester 3 : C10
Semester 4 : C5, C16

-----Press any key to continue-----
```

3. Test Case 3



A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The prompt is 'kali@kali:~/Tucil-2-Stima\$'. The user enters 'python3 main.py'. The script prompts 'enter filename (with extension): tc3.txt'. It then displays four lines of semester data: 'Semester 1 : C5, C7, C3', 'Semester 2 : C55, C8', 'Semester 3 : C2, C10', and 'Semester 4 : C9'. A cursor is visible on the line for Semester 2. At the bottom, it says '-----Press any key to continue-----' followed by a cursor.

```
kali@kali:~/Tucil-2-Stima$ python3 main.py
enter filename (with extension): tc3.txt
Semester 1 : C5, C7, C3
Semester 2 : C55, C8
Semester 3 : C2, C10
Semester 4 : C9

-----Press any key to continue-----
```

4. Test Case 4



A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The prompt is 'kali@kali:~/Tucil-2-Stima\$'. The user enters 'python3 main.py'. The script prompts 'enter filename (with extension): tc4.txt'. It then displays five lines of semester data: 'Semester 1 : C3', 'Semester 2 : C1', 'Semester 3 : C4', 'Semester 4 : C2', and 'Semester 5 : C5'. A cursor is visible on the line for Semester 3. At the bottom, it says '-----Press any key to continue-----' followed by a cursor.

```
kali@kali:~/Tucil-2-Stima$ python3 main.py
enter filename (with extension): tc4.txt
Semester 1 : C3
Semester 2 : C1
Semester 3 : C4
Semester 4 : C2
Semester 5 : C5

-----Press any key to continue-----
```

5. Test Case 5



A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The prompt is 'kali@kali:~/Tucil-2-Stima\$'. The user enters 'python3 main.py'. The script prompts 'enter filename (with extension): tc5.txt'. It then displays the following data:

```
Semester 1 : C17
Semester 2 : C0, C5, C6
Semester 3 : C2, C20, C4
Semester 4 : C1
```

The cursor is positioned at the end of the fourth line. Below the data, the text '-----Press any key to continue-----' is displayed with a cursor at the end.

6. Test Case 6

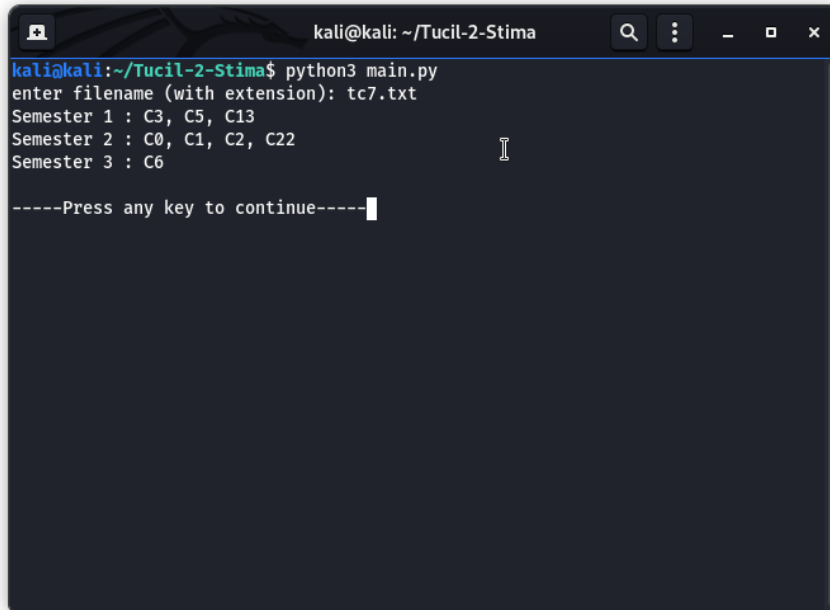


A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The prompt is 'kali@kali:~/Tucil-2-Stima\$'. The user enters 'python3 main.py'. The script prompts 'enter filename (with extension): tc6.txt'. It then displays the following data:

```
Semester 1 : C4, C5
Semester 2 : C0, C1, C66
Semester 3 : C3
```

The cursor is positioned at the end of the third line. Below the data, the text '-----Press any key to continue-----' is displayed with a cursor at the end.

7. Test Case 7




A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The user enters 'tc7.txt' as the filename. The script outputs the following course numbers for each semester: Semester 1: C3, C5, C13; Semester 2: C0, C1, C2, C22; Semester 3: C6. The prompt '-----Press any key to continue-----' is displayed at the bottom.

```
kali@kali:~/Tucil-2-Stima$ python3 main.py
enter filename (with extension): tc7.txt
Semester 1 : C3, C5, C13
Semester 2 : C0, C1, C2, C22
Semester 3 : C6

-----Press any key to continue-----
```

8. Test Case 8



A terminal window titled 'kali@kali: ~/Tucil-2-Stima' showing the execution of a Python script. The user enters 'tc8.txt' as the filename. The script outputs the following course numbers for each semester: Semester 1: C1; Semester 2: C19; Semester 3: C3, C5; Semester 4: C4. The prompt '-----Press any key to continue-----' is displayed at the bottom.

```
kali@kali:~/Tucil-2-Stima$ python3 main.py
enter filename (with extension): tc8.txt
Semester 1 : C1
Semester 2 : C19
Semester 3 : C3, C5
Semester 4 : C4

-----Press any key to continue-----
```


D. Alamat *Source Code*

1. [Disini](#)
2. <https://github.com/Uyamikun/Tucil-2-Stima> (Private Repo)

E. Kesimpulan

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.	✓	