

**Laporan Tugas Kecil 2 IF2211 Strategi Algoritma
Penyusunan Rencana Kuliah dengan *Topological Sort*
(Penerapan *Decrease and Conquer*)**

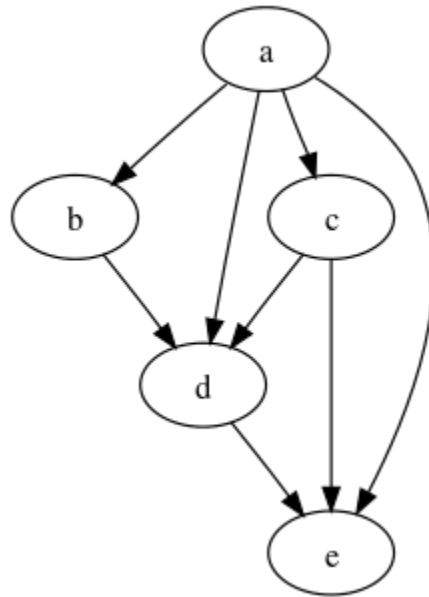


(Nama - NIM)
Girvin Junod - 13519096

Semester II Tahun 2020/2021

I. Algoritma Topological Sort dan kaitannya dengan strategi Decrease and Conquer

Topological sort adalah pengurutan node dari suatu *Directed Acyclic Graphs* (DAG) sehingga untuk setiap sisi uv dari node u ke node v , u selalu ada sebelum v dalam pengurutannya. Dalam aplikasinya, DAG dapat merepresentasikan daftar tugas yang perlu dilakukan dan tugas yang perlu diselesaikan untuk dapat melakukan tugas yang lain, *topological sorting* dalam hal ini akan memberi salah satu urutan pengerjaan tugas yang dapat dilakukan. *Topological sort* hanya dapat dilakukan jika dan hanya jika graf berupa DAG, yaitu graf berarah dan asiklik. Setiap DAG juga pasti memiliki *topological order* minimal satu.



Contoh *Directed Acyclic Graph*

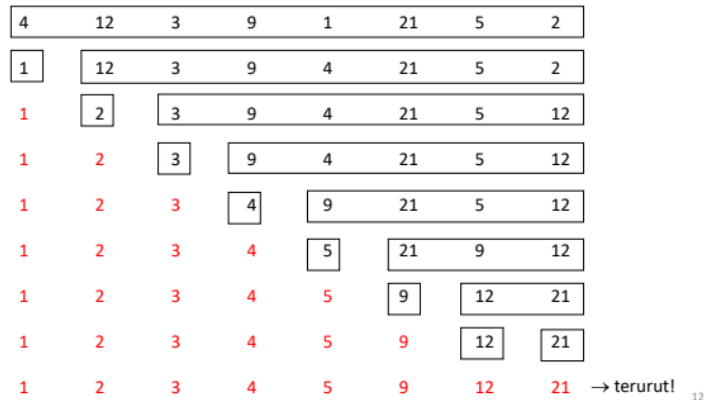
Sumber: en.wikipedia.org/wiki/Directed_acyclic_graph

Strategi *decrease and conquer* sendiri adalah strategi algoritma yang mereduksi persoalan menjadi beberapa persoalan yang lebih kecil (upa-persoalan) lalu hanya memproses satu upa-persoalan saja secara rekursif untuk mencari solusinya. Biasanya upa-persoalan yang dihasilkan hanya dua saja. Contoh penerapan strategi *decrease and conquer* adalah untuk *selection sort*. Dalam *selection sort*, suatu larik dibagi menjadi dua upalarik dengan satu upalarik hanya berisi satu elemen saja. Upalarik dengan satu elemen ini berisi nilai minimum dari larik. Lalu dipilih upalarik yang lainnya untuk diproses lagi secara rekursif dimana pembagian menjadi dua upalarik ini akan dilakukan terus menerus sampai tersisa 2 elemen untuk dibagi menjadi dua upalarik dengan satu elemen. Elemen minimum pada iterasi ke- i yaitu elemen pada upalarik dengan satu elemen ini ditempatkan pada larik posisi ke- i sehingga pada akhirnya akan didapatkan suatu larik yang terurut dari kecil ke besar. Untuk mengubah urutan menjadi dari besar ke kecil hanya perlu mengubah elemen dari upalarik dengan satu elemen menjadi elemen dengan nilai maksimal.

Contoh 1. Misalkan tabel A berisi elemen-elemen berikut:

4 12 3 9 1 21 5 2

Langkah-langkah pengurutan dengan *Selection Sort*:



Ilustrasi *selection sort* dengan strategi *decrease and conquer*

Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/stima20-21.htm

Algoritma *topological sort* memiliki kaitan dengan strategi *decrease and conquer* karena strategi tersebut juga dipakai dalam implementasi *topological sort*. Pertama, dicari node-node dalam DAG yang tidak memiliki sisi yang berakhir di node itu. Lalu node-node tersebut dihapus dari DAG dan disimpan sebagai urutan node hasil *topological sort* untuk iterasi pertama. Proses ini dilakukan lagi dengan DAG sisa dan seterusnya sampai tidak ada node lagi di DAG. Ini adalah implementasi dari strategi *decrease and conquer*, dengan DAG dibagi menjadi dua upa-persoalan tiap iterasi yaitu upa-persoalan berisi node yang tidak memiliki sisi yang berakhir di node itu dan upa-persoalan berisi node sisa yang akan diproses untuk iterasi selanjutnya. Urutan node hasil *topological sort* didapat dari node yang dihapus dari tiap iterasi dengan urutan dari iterasi awal sampai akhir.

Dalam tugas ini, *topological sort* digunakan untuk menentukan mata kuliah apa saja yang dapat diambil seorang mahasiswa tiap semester dari sebuah daftar mata kuliah. Jadi mata kuliah beserta mata kuliah prasyarat pengambilannya di sini merupakan sebuah DAG dengan tiap node berupa mata kuliah dan tiap sisi uv berarti mata kuliah u adalah prasyarat untuk mengambil mata kuliah v. Dengan *topological sort*, akan didapat suatu urutan mata kuliah yang bisa diambil tiap semester sampai dengan semester 8.

Dengan daftar mata kuliah sebagai DAG, mata kuliah sebagai node, dan tiap sisi uv yang menandakan mata kuliah u merupakan prasyarat pengambilan mata kuliah v, persoalan ini dapat diselesaikan sama seperti penyelesaian *topological sort* biasa dengan strategi *decrease and conquer*. Untuk penyelesaiannya, diambil mata kuliah yang prasyarat pengambilannya sudah terpenuhi dari daftar mata kuliah lalu disimpan sebagai mata kuliah yang bisa diambil untuk semester pertama. Proses ini diulang untuk semester selanjutnya dengan daftar mata kuliah yang sudah berkurang dan diulang terus sampai tidak ada lagi mata kuliah yang tersisa di daftar mata kuliah atau sudah melewati semester 8.

II. Source Code (Python)

Kode program dibagi menjadi penulisan nama aplikasi, pembacaan file input, topological sorting, output, dan main program.

```
1  #Girvin Junod 13519096 K02 Tupil Stima 2
2
3  #Program ini terdiri dari fungsi baca soal dari file, toposortnya, dan output
4  > def namaaplikasi(): #Print nama aplikasi...
5
6
7  > def bacafile(namafile): #membaca input dari file, output list representasi DAG sesuai format...
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34 > def toposort(listdag, hasil): ...
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67 > def outputkuliah(hasil): #input list yang sudah tersort, output menuliskan hasil sesuai tujuan program yaitu untuk mengambil mata kuliah...
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98 #syarat, soal harus ada di folder yang sama dengan file python
99
100 #Main Program
101
102 import os
103 from pathlib import Path
104 path = os.path.dirname(Path(__file__).absolute().parent)
105
106 namafile = "soal2.txt" #ganti nama file disini
107
108 namafile1 = os.path.join(path, 'test', namafile)
109 temp = []
110 namaaplikasi()
111 listkuliah = bacafile(namafile1)
112 hasil = toposort(listkuliah, temp)
113 outputkuliah(hasil)
```

Main program

```
#Girvin Junod 13519096 K02 Tupil Stima 2

#Program ini terdiri dari fungsi baca soal dari file, toposortnya, dan output
def namaaplikasi(): #Print nama aplikasi
    print("SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:")
```

Penulisan nama aplikasi

```
7  def bacafile(namafile): #membaca input dari file, output list representasi DAG sesuai format
8      ...
9      Format penulisan soal seperti
10     A, B, C, D.
11     E, F.
12     G.
13     H, I.
14     ...
15     #soal asumsi ada di folder yang sama dengan file python
16     f = open(namafile) #baca soal
17     read = f.read().split('\n') #dipisahkan dari newline
18     listkuliah = [] #list representasi DAG. List berisi list lainnya.
19     for i in range(len(read)):
20         read[i] = read[i].replace('.', '') #menghilangkan titik
21         if read[i] != '': #jika bukan empty space
22             templist = read[i].split(',') #dipisahkan berdasarkan koma
23             for j in range(len(templist)): #Menghapus trailing dan leading white space
24                 templist[j] = templist[j].strip()
25             kosong = False
26             for k in range(len(templist)): #Pengecekan jika ada yang list kosong
27                 if templist[k] == '':
28                     kosong = True
29                     break
30             if kosong == False:
31                 listkuliah.append(templist) #Dimasukkan ke DAG
32     return listkuliah
33
```

Pembacaan input dari file

```

34 def toposort(listdag, hasil):
35     #Untuk toposort, input list representasi DAG sesuai format, output list berisi node yang sudah di topo sort
36     #Node tingkat sama (sama-sama bisa diambil) dikelompokkan menjadi satu string dan dipisah dengan koma
37     #Contoh output: ['A', 'B', 'C', 'D']
38     if len(listdag) != 0: #Selama DAG masih berisi
39         now = [] #list sementara penyimpan node yang bisa diambil untuk semester ini
40         keambil = False
41         for j in listdag:
42             if len(j) == 1: #Mengambil semua node yang syaratnya sudah terpenuhi
43                 keambil = True
44                 ambil = j[0] #Karena tinggal 1 ambil elemen pertama
45                 now.append(ambil) #Dimasukkan ke list sementara
46         if keambil:
47             k = 0
48             while k < len(listdag):
49                 for l in now:
50                     if l in listdag[k]: #Menghapus node yang diambil dari syarat node lain
51                         listdag[k].remove(l)
52                     if len(listdag[k]) == 0: #Menghapus node yang diambil dari list representasi DAG
53                         listdag.remove(listdag[k])
54                         k-=1
55                 k+=1
56         #output
57         str = ""
58         for n in range (len(now)):
59             if n == len(now) - 1: #Formatting output
60                 str += now[n]
61             else:
62                 str += now[n]
63                 str += ", "
64         hasil.append(str)
65         toposort(listdag,hasil) #rekursi dengan dag yg lebih kecil
66     return hasil

```

Topological Sort

```

68 def outputkuliah(hasil): #input list yang sudah tersort, output menuliskan hasil sesuai tujuan program yaitu untuk mengambil mata kuliah
69     #Format output menurut persoalan mata kuliah
70     for i in range (len(hasil)):
71         if i > 7: #Kalau lebih dari 8 semester dipotong
72             break
73         #Merubah angka ke angka romawi untuk output
74         if i+1 == 1:
75             roman = "I"
76         elif i+1 == 2:
77             roman = "II"
78         elif i+1 == 3:
79             roman = "III"
80         elif i+1 == 4:
81             roman = "IV"
82         elif i+1 == 5:
83             roman = "V"
84         elif i+1 == 6:
85             roman = "VI"
86         elif i+1 == 7:
87             roman = "VII"
88         elif i+1 == 8:
89             roman = "VIII"
90
91         str = "Semester " + roman + ": "
92         print(str + hasil[i], end= " ")
93         if i == len(hasil) - 1: #Kalau yang paling akhir kasih titik di akhirnya
94             print(".")
95         else:
96             print()

```

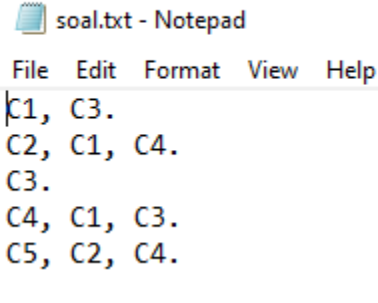
Penulisan Output

III. Tangkapan Layar dan Input dan Output

1.

Input:

```
#ganti nama soal di sini
namafile = "soal.txt"
```



soal.txt - Notepad

File Edit Format View Help

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

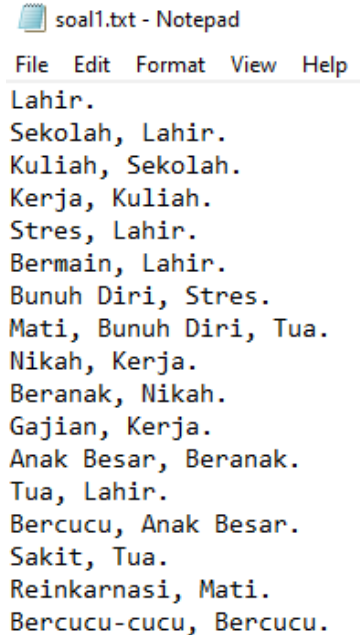
Output:

```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: C3
Semester II: C1
Semester III: C4
Semester IV: C2
Semester V: C5.
PS D:\Coding\tucilstima2> |
```

2.

Input:

```
#ganti nama soal di sini
namafile = "soal1.txt"
```



soal1.txt - Notepad

File Edit Format View Help

Lahir.
Sekolah, Lahir.
Kuliah, Sekolah.
Kerja, Kuliah.
Stres, Lahir.
Bermain, Lahir.
Bunuh Diri, Stres.
Mati, Bunuh Diri, Tua.
Nikah, Kerja.
Beranak, Nikah.
Gajian, Kerja.
Anak Besar, Beranak.
Tua, Lahir.
Bercucu, Anak Besar.
Sakit, Tua.
Reinkarnasi, Mati.
Bercucu-cucu, Bercucu.

Output:

```

PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: Lahir
Semester II: Sekolah, Stres, Bermain, Tua
Semester III: Kuliah, Bunuh Diri, Sakit
Semester IV: Kerja, Mati
Semester V: Nikah, Gajian, Reinkarnasi
Semester VI: Beranak
Semester VII: Anak Besar
Semester VIII: Bercucu
PS D:\Coding\tucilstima2>

```

3.

Input:

```

namafile = "soal2.txt"
soal2.txt - Notepad
File Edit Format View Help
MA1101.
FI1101.
KU1001.
KU1102.
KU1011.
KU1024.

MA1201, MA1101.
FI1201, FI1101.
IF1210, KU1102.
KU1202, KU1102.
KI1002, KU1011.
EL1200, FI1101.

IF2121, IF1210, MA1101, MA1201.
IF2110, KU1102, IF1210.
IF2120, MA1201, MA1101.
IF2124, EL1200.
IF2123, MA1201.
IF2130, KU1202.

IF2210, IF2110.
IF2211, IF2110.
IF2220, MA1101, MA1201, IF2120.
IF2230, IF2130.
IF2240, IF2121, IF2120.
IF2250, KU1202, IF2110.

IF3170, IF2121, IF2124, IF2220, IF2211.
IF3110, IF2210, IF2110.
IF3130, IF2230.
IF3141, IF2240, IF2250.
IF3150, IF2250.
IF3140, IF2240.
IF3151, IF2250.

IF3210, IF2110, IF2130, IF3110.
IF3270, IF2210, IF3170.
IF3230, IF3130.
IF3250, IF2250, IF3150.
IF3260, IF2123, IF2110, IF2130, IF3151.
IF3280, IF3151, IF3150.

IF4090, IF3280.
IF4091, IF3280.

IF4092, IF4091.

```

Output:

```

PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: MA1101, FI1101, KU1001, KU1102, KU1011, KU1024
Semester II: MA1201, FI1201, IF1210, KU1202, KI1002, EL1200
Semester III: IF2121, IF2110, IF2120, IF2124, IF2123, IF2130
Semester IV: IF2210, IF2211, IF2220, IF2230, IF2240, IF2250
Semester V: IF3170, IF3110, IF3130, IF3141, IF3150, IF3140, IF3151
Semester VI: IF3210, IF3270, IF3230, IF3250, IF3260, IF3280
Semester VII: IF4090, IF4091
Semester VIII: IF4092.
PS D:\Coding\tucilstima2>

```

4.

Input:

```

namafile = "soal3.txt"
soal3.txt - Notepad
File Edit Format View Help
Machine Learning, AI, Algoritma Struktur Data.
AI, Logika Komputasional, Teori Bahasa Formal dan Otomata, Probabilitas Statistika, Strategi Algoritma.
Probabilitas Statistika, Matematika Diskrit.
Strategi Algoritma.
Matematika Diskrit.
Algoritma Struktur Data.
Logika Komputasional.
Teori Bahasa Formal dan Otomata.

```

Output:

```

PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: Strategi Algoritma, Matematika Diskrit, Algoritma Struktur Data, Logika Komputasional, Teori Bahasa Formal dan Otomata
Semester II: Probabilitas Statistika
Semester III: AI
Semester IV: Machine Learning.
PS D:\Coding\tucilstima2>

```

5.

Input:

```

namafile = "soal4.txt"
soal4.txt - Notepad
File Edit Format View Help
Kriptografi, Matdis.

```

Kalkulus.

TBFO, Matdis.

Fisika.

Stima, Matdis, Kalkulus.

Matdis, Kalkulus.

Output:


```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: Kalkulus, Fisika
Semester II: Matdis
Semester III: Kriptografi, TBFO, Stima.
PS D:\Coding\tucilstima2>
```

6.

Input:

```
namafile = "soal5.txt"
```

soal5.txt - Notepad

File Edit Format View

A.
B.
C.
D.
E.
F.
G.
H.
I.
J.
K, A.
L, K.
M, L.
N, M.
O, N.
P, O.
Q, P.
R, Q.
S, R.
T, S.
U, T.
V, U.
W, V.
X, W.
Y, X.
Z, Y.

Output:

```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: A, B, C, D, E, F, G, H, I, J
Semester II: K
Semester III: L
Semester IV: M
Semester V: N
Semester VI: O
Semester VII: P
Semester VIII: Q
PS D:\Coding\tucilstima2>
```

7.

Input:

```
namafile = "soal6.txt"
soal6.txt - Notepad
File Edit Format View Help
Forgotten Crossroads.
Vengeful Spirit, Forgotten Crossroads.
Greenpath, Vengeful Spirit.
Mothwing Cloak, Greenpath.
City of Tears, Mothwing Cloak.
Ancient Basin, City of Tears.
Soul Sanctum, City of Tears.
Desolate Dive, Soul Sanctum.
Crystal Peak, Desolate Dive.
Resting Grounds, Crystal Peak.
Dream Nail, Crystal Peak.
Crystal Heart, Crystal Peak.
Royal Waterways, City of Tears.
Isma's Tear, Crystal Peak, Royal Waterways.
Fungal Waste, Greenpath.
Mantis Claw, Fungal Waste, Mothwing Cloak.
Shade Cloak, The Abyss, Mothwing Cloak.
The Abyss, King's Brand.
Kingdom's Edge, City of Tears.
King's Brand, Kingdom's Edge.
Queen's Garden, Isma's Tear, Fog Canyon.
Fog Canyon, Greenpath.
Monomon the Teacher, Shade Cloak, Fog Canyon.
Deepnest, Fungal Waste.
Herrah the Beast, Deepnest.
White Palace, Dream Nail.
Monarch Wings, Ancient Basin.
Lurien the Watcher, Monarch Wings.
The Hollow Knight, Lurien the Watcher, Herrah the Beast, Monomon the Teacher.
```

Output:

```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: Forgotten Crossroads
Semester II: Vengeful Spirit
Semester III: Greenpath
Semester IV: Mothwing Cloak, Fungal Waste, Fog Canyon
Semester V: City of Tears, Mantis Claw, Deepnest
Semester VI: Ancient Basin, Soul Sanctum, Royal Waterways, Kingdom's Edge, Herrah the Beast
Semester VII: Desolate Dive, King's Brand, Monarch Wings
Semester VIII: Crystal Peak, The Abyss, Lurien the Watcher
PS D:\Coding\tucilstima2>
```

8.

Input:

```
namafile = "soal7.txt"
soal7.txt - Notepad
File Edit Format View Help
Daftar ITB.
Masuk ITB, Daftar ITB.
Keos, Masuk ITB.
Stres, Keos.
Pelajaran Tertinggal, Keos.
Makin Keos, Stres.
Depresi, Stres.
Kapan lulus ya Tuhan, Makin Keos.
```

Output:

```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: Daftar ITB
Semester II: Masuk ITB
Semester III: Keos
Semester IV: Stres, Pelajaran Tertinggal
Semester V: Makin Keos, Depresi
Semester VI: Kapan lulus ya Tuhan.
PS D:\Coding\tucilstima2>
```

9.

Input:

```
namafile = "soal8.txt"
soal8.txt - Notepad
File Edit Format View
```

Output:

```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
PS D:\Coding\tucilstima2>
```

10.

Input:

```
103 namafile = "soal9.txt"
104 soal9.txt - Notepad
105 File Edit Format View Help
106 a.
107
108
```

Output:

```
PS D:\Coding\tucilstima2> python tucil2.py
SUSUNAN RENCANA KULIAH DENGAN APLIKASI SIX*3:
Semester I: a.
PS D:\Coding\tucilstima2>
```

IV. Alamat Drive Tempat Kode:

Google Drive:

<https://drive.google.com/drive/folders/1brbdHILAQfKrasi-p9HSlwfbbvIOJOAK?usp=sharing>

Github:

github.com/girvinjunod/Stima_Tucil2_13519096

V. Checklist Penilaian:

No.	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.	✓	