

LAPORAN TUGAS KECIL II
PENYUSUNAN RENCANA KULIAH DENGAN *TOPOLOGICAL SORT*
(PENERAPAN *DECREASE AND CONQUER*)
IF2211 Strategi Algoritma



Oleh :

Harith Fakhiri Setiawan - 13519161

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021

BAB I

Algoritma *Topological Sort* dan Kaitannya dengan Pendekatan *Decrease and Conquer*

Secara mendasar, *Decrease and Conquer* merupakan metode perancangan algoritma dengan mereduksi persoalan menjadi dua sub-persoalan (sub-problem) yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja.

Sementara *Topological Sort*, merupakan pengurutan simpul dari sebuah *Directed Acyclic Graph* (DAG). Setiap simpul memiliki hubungan satu sama lain yang digambarkan dengan busur. Salah satu pendekatan pada *Topological Sort*, adalah dengan memisahkan atau mengeliminasi simpul yang memiliki derajat nol, kemudian menyelesaikan simpul yang tidak berderajat nol dengan proses rekursif. Kemudian, setiap simpul yang dipisahkan atau dieliminasi akan dijadikan sebagai solusi secara berurut.

Hubungan dari *Topological Sort* dengan pendekatan *Decrease and Conquer* adalah penyelesaian yang sama sama membagi persoalan menjadi sub-persoalan dimana sebuah elemen memiliki derajat masuk bernilai nol, atau derajat masuk lebih dari nol. Apabila derajat masuk bernilai nol, maka elemen akan dipisah dari sub-persoalan baru. Sementara apabila derajat masuk lebih besar dari nol, maka akan dilakukan rekursi pada sub-persoalan untuk mencari solusi hingga tidak ada lagi elemen yang tidak berderajat masuk nol. Pada permasalahan penyusunan mata kuliah ini, mata kuliah yang sudah tidak memiliki *prerequisite* atau berderajat masuk nol akan dipisahkan dari sub-persoalan, kemudian dilakukan rekursi untuk mata kuliah yang sudah masih memiliki *prerequisite* atau berderajat masuk lebih dari nol sehingga semua mata kuliah tidak memiliki derajat masuk atau tidak memiliki *prerequisite*.

Langkah-langkah pengerjaan algoritma :

1. Persoalan diasumsikan selalu berbentuk *Directed Acyclic Graph* (DAG) sehingga selalu dapat diselesaikan dengan Algoritma *Topological Sort* yang dibuat.
2. Untuk menjalankan program, terminal dibuka pada *directory* src, lalu dijalankan program main. Ketika program sudah berjalan, pengguna akan diminta untuk memberi masukan berupa nama file dari persoalan disertai formatnya, contoh: 'test1.txt'
3. Setelah itu, program akan mengarahkan ke *directory* lokasi file, membukanya, lalu membacanya. Isi file tersebut kemudian dari proses, dengan menghilangkan character yang diperlukan '\n', ',', dan '.'. Lalu dihasilkan *array* of string mata kuliah beserta *prerequisite*-nya pada setiap barisnya dari file input..
4. Setelah itu, *array* diproses lebih lanjut sehingga menghasilkan *array of array* of mata kuliah yang berisi mata kuliah beserta *prerequisite*-nya.
5. Setelah itu, *array of array* of mata kuliah digunakan untuk mencari *array* of *array* yang berisi setiap mata kuliah beserta derajat masuk yang dimilikinya atau *prerequisite* nya.
6. Setelah itu, dibuat *array* kosong sebagai inisiasi yang akan diisi oleh solusi yang diselesaikan melalui *Topological Sort*.
7. *Topological Sort* dilakukan dengan melakukan pemeriksaan terlebih dahulu mata kuliah yang tidak memiliki *prerequisite* dengan memeriksa derajat masuk yang bernilai nol. Kemudian, mata kuliah yang bernilai nol akan dimasukkan ke dalam sebuah *array of array* of mata kuliah dari masing masing semester yang pada awal diinisiasi sebagai *array* kosong melalui metode *append*. Kemudian, mata kuliah tersebut dipisahkan atau dieliminasi dari *array of array* of mata kuliah yang berisi berbagai mata kuliah beserta *prerequisite*-nya. Kemudian, dilakukan *update* atau

pembaruan *array of array* yang berisi setiap mata kuliah beserta derajat masuk yang dimilikinya atau *prerequisite* nya menggunakan *array of array* mata kuliah yang telah dieliminasi. Kemudian, dilakukan lagi pencarian mata kuliah lainnya yang juga memiliki nol dengan proses yang sama dan dilakukan *update* atau pembaruan kembali secara terus menerus dengan memanfaatkan proses rekursif. Pada setiap proses rekursif, mata kuliah yang dipisahkan atau dieliminasi dimasukkan ke dalam *array* solusi menggunakan metode *append*. Proses rekursif ini dilakukan hingga semua mata kuliah sudah memiliki derajat masuk nol atau sudah berada pada *array* solusi.

8. Dari *array of array of* solusi mata kuliah tiap semester, dilakukan prosedur untuk menghasilkan *output* sesuai format yang diminta pada layar. Format tersebut berbentuk seperti:

Semester 1 : ID1.

Semester 2 : ID2.

Semester 3 : ID3.

Semester 4 : ID4.

Semester 5 : ID5, ID6, ID7.

Semester 6 : ID8.

BAB II

Source Code

procToArr_13519161.py

```
src > procToArr_13519161.py
1  #fungsi yang menerima input nama file kemudian mengembalikan
2  #array of string perlines dari file input
3  def strmatkul(name):
4      f = open(name,'r')
5      file = f.readlines()
6      operand = []
7      #melakukan replace beberapa character seperti koma, titik, dan newline
8      for lines in file:
9          lines = lines.replace(","," ")
10         lines = lines.replace("\n"," ")
11         lines = lines.replace(".", " ")
12         operand.append(lines)
13     return operand
14
15 #fungsi yang menerima input array of string kemudian
16 #mengembalikan array of array dari elemen pada matkul prereq
17 def makeArr(arr):
18     a = ['' for i in range(len(arr))]
19     i = 0
20     #melakukan split dari string untuk menghasilkan array
21     for lines in arr:
22         #melakukan split dari string untuk menghasilkan array
23         a[i] = lines.split()
24         i+=1
25     return a
26
27 #fungsi yang menerima input berupa array of array dari
28 #elemen matkul prereq, kemudian menghasilkan array of array
29 #yang berisi elemen maktul beserta derajat masuk
30 def makeArrdegreein(arr):
31     a = [['' for j in range(2)] for i in range(len(arr))]
32     i = 0
33     for lines in arr:
34         count = len(arr[i])-1
35         a[i][0] = lines[0] #memasukkan elemen
36         a[i][1] = count #memasukkan derajat masuk
37         i+=1
38     return a
39
```

arrayToTopo_13519161.py

```
src > ArrayToTopo_13519161.py
1  from procToArr_13519161 import strmatkul,makeArr,makeArrdegreein
2
3  #fungsi ini menerima input berupa array of array dari
4  #elemen matkul elemen, dan juga string berupa elemen,
5  #kemudian mengembalikan array of array dengan elemen sudah dihapus
6  def DeleteEl(arr, element):
7      a = ['' for i in range(len(arr)) ]
8      i = 0
9      #melakukan deleting elemen
10     for lines in arr:
11         #merubah array of words menjadi string
12         lines = ' '.join(lines)
13         #me-replace element dengan spasi apabila terdapat elemen pada string
14         lines = lines.replace(element," ")
15         #melakukan split untuk menghasilkan array dari string
16         lines = lines.split()
17         #maemasukkannya ke array a
18         a[i] = lines
19         i+=1
20     a.remove('') #menghapus list kosong
21     return a
22
23 #fungsi ini menerima input berupa array of array dari elemen matkul prereq
24 #array of array dari array derajat masuk, beserta arrFin yang berupa
25 # array yang nantinya akan dikembalikan sebagai solusi topological sort
26 def Topological(arrPre, arrdegree, arrFin):
27     if (len(arrdegree)==0): #apabila sudah tidak terdapat elemen //basis
28         return arrFin
29     else: #rekursif
30         element = []
31         for i in range(len(arrdegree)) :
32             if (arrdegree[i][1] == 0) :
33                 #append elemen yang berderajat 0
34                 element.append(arrdegree[i][0])
35         arrfinnew = []
36         arrFin.append(element)
37         arrfinnew = arrFin
38         #malakukan update terhadap array of array prereq
39         newDelpre = arrPre
40         for words in element:
41             #menghapus elemen yang berderajat 0
42             newDelpre = DeleteEl(newDelpre,words)
43         #melakukan update terhadap array derajat masuk
44         decdegree = makeArrdegreein(newDelpre)
45         return Topological(newDelpre,decdegree,arrfinnew)
```

```

46
47 #prosedur ini menerima array of array elemen matkul yang
48 #kemudian melakukan print berupa output sesuai dengan format
49 #yang dibuat
50 def outputSemester(arrayfinal):
51     i = 1
52     #malakukan printing solusi
53     for semester in arrayfinal:
54         print("Semester ",i," : ",end="")
55         for j in range(len(semester)):
56             print(semester[j],end="")
57             if (j==len(semester)-1) :
58                 print(".",end="")
59             else:
60                 print(", ",end="")
61         print()
62         i+=1

```

main_13519161.py

```

src > main_13519161.py
1 from ArrayToTopo_13519161 import DeleteEl,Topological,outputSemester
2 from procToArr_13519161 import strmatkul,makeArr,makeArrdegreein
3
4 def main():
5     filename = str(input("Masukkan nama file prereq mata kuliah: "))
6     #memasukkan input nama file dengan format nya, contoh "<namafile>.txt"
7     fname = "../test/"+filename
8     #membuat array perlines dari input
9     a = strmatkul(fname)
10    #membuat array of array of elemen matkul dari prereq input
11    arrayPre = makeArr(a)
12    #membuat array of array yang berisi elemen dan jumlah derajat masuk
13    arraydegree = makeArrdegreein(arrayPre)
14    sort=[] #inisiasi array kosong
15    # melakukan topological sorting
16    topo = Topological(arrayPre,arraydegree,sort)
17    print()
18    print("Solusi penyusunan rencana kuliah :")
19    print()
20    #output print solusi
21    outputSemester(topo)
22    print()
23
24
25 main()

```

BAB III

Uji Kasus

1. Test 1

Input:

```
test > ≡ test1.txt
1    C1, C3.
2    C2, C1, C4.
3    C3.
4    C4, C1, C3.
5    C5, C2, C4.
```

Output:

```
Masukkan nama file prereq mata kuliah: test1.txt

Solusi penyusunan rencana kuliah :

Semester 1 : C3.
Semester 2 : C1.
Semester 3 : C4.
Semester 4 : C2.
Semester 5 : C5.
```

2. Test 2

Input:

```
test > ≡ test2.txt
1    IF01, IF03, IF05.
2    IF02, IF03.
3    IF03.
4    IF04, IF01, IF02, IF05.
5    IF05.
6    IF06, IF05.
7    IF07, IF04.
8    IF08, IF04.
9    IF09, IF07, IF08.
10   IF10, IF06, IF08, IF11.
11   IF11, IF06.
12   IF12, IF09, IF10.
13   IF13, IF10.
```

Output:

```
Masukkan nama file prereq mata kuliah: test2.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : IF03, IF05.  
Semester 2 : IF01, IF02, IF06.  
Semester 3 : IF04, IF11.  
Semester 4 : IF07, IF08.  
Semester 5 : IF09, IF10.  
Semester 6 : IF12, IF13.
```

3. Test 3

Input:

```
test > ≡ test3.txt
```

```
1 ID1.  
2 ID2, ID1.  
3 ID3, ID1, ID2.  
4 ID4, ID2, ID3.  
5 ID5, ID2, ID4.  
6 ID6, ID4.  
7 ID7, ID3, ID4.  
8 ID8, ID5, ID6, ID7.
```

Output:

```
Masukkan nama file prereq mata kuliah: test3.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : ID1.  
Semester 2 : ID2.  
Semester 3 : ID3.  
Semester 4 : ID4.  
Semester 5 : ID5, ID6, ID7.  
Semester 6 : ID8.
```


4. Test 4

Input:

```
test > ≡ test4.txt
1   Kalkulus_IA.
2   Pengenalan_Komputasi.
3   Kalkulus_IIA,Kalkulus_IA.
4   Dasar_Pemrograman,Pengenalan_Komputasi.
5   Algoritma_Struktur_Data, Dasar_Pemrograman.
6   Pemrograman_Berbasis_Objek, Algoritma_Struktur_Data.
7   Basis_Data, Logika_Komputasional, Dasar_Pemrograman.
8   Logika_Komputasional,Kalkulus_IIA,Pengenalan_Komputasi.
```

Output:

```
Masukkan nama file prereq mata kuliah: test4.txt

Solusi penyusunan rencana kuliah :

Semester 1 : Kalkulus_IA, Pengenalan_Komputasi.
Semester 2 : Kalkulus_IIA, Dasar_Pemrograman.
Semester 3 : Algoritma_Struktur_Data, Logika_Komputasional.
Semester 4 : Pemrograman_Berbasis_Objek, Basis_Data.
```

5. Test 5

Input:

```
test > ≡ test5.txt
1   MA1101.
2   FI1101.
3   KU1001.
4   KU1102.
5   KU1011.
6   KU1024.
7   MA1201, MA1101.
8   FI1201, FI1101.
9   IF1210, KU1102.
10  KU1202, KU1102.
```

11 KI1002, KU1011.
12 EL1200, FI1101.
13 IF2121, IF1210, MA1101, MA1201.
14 IF2110, KU1102, IF1210.
15 IF2120, MA1201, MA1101.
16 IF2124, EL1200.
17 IF2123, MA1201.
18 IF2130, KU1202.
19 IF2210, IF2110.
20 IF2211, IF2110.

20 IF2211, IF2110.
21 IF2220, MA1101, MA1201, IF2120.
22 IF2230, IF2130.
23 IF2240, IF2121, IF2120.
24 IF2250, KU1202, IF2110.
25 IF3170, IF2121, IF2124, IF2220, IF2211.
26 IF3110, IF2210, IF2110.
27 IF3130, IF2230.
28 IF3141, IF2240, IF2250.
29 IF3150, IF2250.
30 IF3140, IF2240.
31 IF3151, IF2250.
32 IF3210, IF2110, IF2130, IF3110.
33 IF3270, IF2210, IF3170.
34 IF3230, IF3130.
35 IF3250, IF2250, IF3150.
36 IF3260, IF2123, IF2110, IF2130, IF3151.
37 IF3280, IF3151, IF3150.
38 IF4090, IF3280.
39 IF4091, IF3280.
40 IF4092, IF4091.

Output:

```
Masukkan nama file prereq mata kuliah: test5.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : MA1101, FI1101, KU1001, KU1102, KU1011, KU1024.  
Semester 2 : MA1201, FI1201, IF1210, KU1202, KI1002, EL1200.  
Semester 3 : IF2121, IF2110, IF2120, IF2124, IF2123, IF2130.  
Semester 4 : IF2210, IF2211, IF2220, IF2230, IF2240, IF2250.  
Semester 5 : IF3170, IF3110, IF3130, IF3141, IF3150, IF3140, IF3151.  
Semester 6 : IF3210, IF3270, IF3230, IF3250, IF3260, IF3280.  
Semester 7 : IF4090, IF4091.  
Semester 8 : IF4092.
```

6. Test 6

Input:

```
test > ≡ test6.txt
```

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

Output:

```
Masukkan nama file prereq mata kuliah: test6.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : C3, C5.  
Semester 2 : C1, C6.  
Semester 3 : C4.  
Semester 4 : C7.  
Semester 5 : C2.
```

7. Test 7

Input:

```
test > ≡ test7.txt
1  Fisika_IA.
2  Kalkulus_IA.
3  Fisika_IIA, Fisika_IA.
4  Kalkulus_IIA, Kalkulus_IA.
5  PAR, Kalkulus_IIA, Fisika_IIA.
6  Kimia_IIA, PAR, DASPRO.
7  PRD, Kalkulus_IIA.
8  DASPRO, PRD.
```

Output:

```
Masukkan nama file prereq mata kuliah: test7.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : Fisika_IA, Kalkulus_IA.
Semester 2 : Fisika_IIA, Kalkulus_IIA.
Semester 3 : PAR, PRD.
Semester 4 : DASPRO.
Semester 5 : Kimia_IIA.
```

8. Test 8

Input:

```
test > ≡ test8.txt
1  Fisika_IA.
2  Kalkulus_IA.
3  Pengenalan_Komputasi.
4  Fisika_IIA, Fisika_IA.
5  Kalkulus_IIA, Kalkulus_IA.
6  Statistika_Industri, Kalkulus_IIA.
7  Simulasi_Komputer, Pengenalan_Komputasi.
8  Ekonomi_Teknik, Statistika_Industri.
9  Proses_Manufaktur, Kalkulus_IIA.
10 Sistem_Manufaktur_Terintegrasi_Komputer, Simulasi_Komputer, Proses_Manufaktur.
```

Output:

```
Masukkan nama file prereq mata kuliah: test8.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : Fisika_IA, Kalkulus_IA, Pengenalan_Komputasi.  
Semester 2 : Fisika_IIA, Kalkulus_IIA, Simulasi_Komputer.  
Semester 3 : Statistika_Industri, Proses_Manufaktur.  
Semester 4 : Ekonomi_Teknik, Sistem_Manufaktur_Terintegrasi_Komputer.
```

9. Test 9

Input:

```
test > ≡ test9.txt
```

```
1 FIS1A, MAT1A.  
2 MAT1A.  
3 KIM1A, MAT1A.  
4 BIO1A, FIS1A, KIM1A.
```

Output:

```
Masukkan nama file prereq mata kuliah: test9.txt
```

```
Solusi penyusunan rencana kuliah :
```

```
Semester 1 : MAT1A.  
Semester 2 : FIS1A, KIM1A.  
Semester 3 : BIO1A.
```

10. Test 10

Input :

```
test > ≡ test10.txt
1      E0.
2      E1, E0.
3      E2, E0.
4      E3, E0, E2.
5      E4, E3, E6.
6      E5, E0, E3.
7      E6, E0, E7.
8      E7, E8.
9      E8.
10     E9, E4, E6.
11     F0, E9.
12     F1, E9.
13     F2, E9, F1.
14     F3, E4, F2.
15     F4, F0, F2.
```

Output :

```
Masukkan nama file prereq mata kuliah: test10.txt

Solusi penyusunan rencana kuliah :

Semester 1 : E0, E8.
Semester 2 : E1, E2, E7.
Semester 3 : E3, E6.
Semester 4 : E4, E5.
Semester 5 : E9.
Semester 6 : F0, F1.
Semester 7 : F2.
Semester 8 : F3, F4.
```

BAB IV

Google drive : <https://drive.google.com/drive/folders/1bLBEK2UxMov5aEV-nHJrrx6nTLApARVr>

Github : <https://github.com/harithfakhiri/Tucil-2-Stima>

BAB V

Tabel Penilaian

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	√	